

FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Positional Bias in Object Detection Models

Rodrigo José Ramos Moucho



Mestrado em Engenharia Informática e Computação

Supervisor: Prof. Jaime dos Santos Cardoso

Second Supervisor: Gonçalo Almeida

Third Supervisor: Ricardo Araújo

July 16, 2026

Positional Bias in Object Detection Models

Rodrigo José Ramos Moucho

Mestrado em Engenharia Informática e Computação

Approved in oral examination by the committee:

President: Prof. Carla Gonçalves

Examiner: Prof. Bisheng Wang

Supervisor: Prof. Jaime Cardoso

July 16, 2026

Resumo

Os detetores de objetos são habitualmente avaliados através de métricas agregadas que podem ocultar falhas espacialmente localizadas ou deteções inconsistentes do mesmo objeto em diferentes posições dentro do enquadramento da imagem. Estas inconsistências podem passar facilmente despercebidas e causar consequências graves, particularmente em aplicações críticas onde um alvo pode surgir em qualquer ponto da imagem.

Esta dissertação estuda as inconsistências de deteção dependentes da posição, utilizando um conjunto de dados de inspeção de defeitos em cortiça, no qual o mesmo defeito físico pode surgir duas vezes na mesma imagem, com uma diferença vertical fixa. Isto cria uma experiência de translação entre um par de objetos, que permite estudar como alvos visualmente idênticos, surgindo em diferentes posições numa única imagem, podem produzir previsões significativamente distintas. Uma pipeline de avaliação desenvolvida quantifica a consistência de deteção entre o par, sendo testadas diferentes estratégias de mitigação da assimetria, demonstrando que a redução do stride da primeira camada convolucional de 2 para 1 produz melhorias significativas na consistência do par, enquanto uma estratégia mista que incorpora a sobreamostragem (*oversampling*) de imagens com defeitos em pares alcança os melhores resultados globais.

Para explorar o mecanismo latente, é conduzido um varrimento controlado de translação em tapete rolante (*treadmill*), revelando e permitindo a quantificação rigorosa de como a confiança do detetor varia sistematicamente segundo uma modulação periódica estruturada para alvos transladados verticalmente. Esta estrutura depende do tamanho da imagem e da resolução de inferência e escala com o *stride* do *backbone*. Ablações arquiteturais confirmam que esta periodicidade tem origem na subamostragem por *stride* (*strided downsampling*) e se propaga através da *feature pyramid*. Experiências adicionais mostram que a composição dos dados de treino determina qual o *stride pathway* que domina as previsões, o que por sua vez determina o período observado. As estratégias de mitigação empregadas conseguem reduzir a variabilidade da confiança dos objetos transladados, mas não eliminam a estrutura periódica.

Estas descobertas estabelecem a subamostragem espacial (*spatial downsampling*) como uma das principais causas arquiteturais tanto das inconsistências posicionais como da variação periódica de confiança observada sob translação.

Palavras-chave: Deteção de objetos • YOLO • equivariância à translação • stride • periodicidade subharmónica • inspeção industrial

Abstract

Object detectors are commonly evaluated using aggregated metrics that can obscure spatially localized failures or inconsistent detection for the same object in different locations within the image frame. These inconsistencies can easily go unnoticed and cause grave consequences, particularly in critical applications where a target may appear anywhere in the image.

This dissertation studies position-dependent detection inconsistencies using a cork defect inspection dataset in which the same physical defect may appear twice in the same image at a fixed vertical offset. This creates a natural pair-like translation experiment that allows a study of how visually identical targets, which appear in different positions within a single image, can produce significantly different predictions. A custom-built evaluation pipeline quantifies detection consistency between the pair, and different asymmetry mitigation strategies are tested, showing that reducing the stride of the first convolutional layer from 2 to 1 yields significant improvements in pair consistency, while a mixed strategy incorporating data oversampling of images with pair defects achieves the best overall results.

To explore the latent mechanism, a controlled treadmill translation sweep is conducted, revealing and enabling rigorous quantification of how detector confidence varies systematically under a structured periodic modulation for vertically translated targets. This structure depends on the image size and inference resolution and scales with the backbone’s stride. Architectural ablations confirm that this periodicity originates from strided downsampling and propagates through the feature pyramid. Further experiments show that the training data composition determines which stride pathway dominates predictions, which in turn governs the observed period. Employed mitigation strategies can reduce the confidence variability of translated objects, but do not eliminate the periodic structure.

These findings establish spatial downsampling as a major architectural cause for both positional inconsistencies and the periodic confidence variation observed under translation.

Keywords: Object detection • YOLO • translation equivariance • stride • sub-harmonic periodicity • industrial inspection

The United Nations Sustainable Development Goals

The United Nations Sustainable Development Goals (SDGs) provide a global framework for achieving a better and more sustainable future for all. It includes 17 goals (<https://sdgs.un.org/goals>) to address the world's most pressing challenges, including poverty, inequality, climate change, environmental degradation, peace, and justice.

The specific Sustainable Development Goals mentioned have the following names:

SDG 9 Build resilient infrastructure, promote inclusive and sustainable industrialization and foster innovation

SDG 12 Ensure sustainable consumption and production patterns

SDG	Target	Contribution	Performance Indicators and Metrics
9	9.4	Improving spatial robustness and detection consistency increases model reliability and avoids position-dependent missed detections, enabling more efficient industrial quality-control processes and better use of resources.	The developed evaluation pipeline quantifies detection inconsistencies, and the dissertation broadens the understanding of the consequences of positional bias.
	9.5	Advancing the scientific understanding of positional bias in object detectors upgrades the technological capability of AI-based industrial inspection.	Reinforcing the importance of adopting translation-sensitivity evaluation in detection pipelines, supported by the visualization of translation experiments.
12	12.5	More consistent defect detection reduces undetected defects, allowing flawed cork to be identified earlier and redirected to alternative uses rather than discovered as an error later on.	Reduction in material waste caused by inspection errors.

Acknowledgements

Firstly, I would like to express my gratitude to my supervisor, Professor Jaime S. Cardoso, for his guidance and insight throughout the course of this dissertation.

I am also grateful to Ricardo Araújo, Tiago Madeira, and Gonçalo Almeida, and to everyone at JTA: The Data Scientists, for their daily support and for welcoming me into the team so effortlessly. Their guidance and willingness to help made an enormous difference in my work, and I am thankful for the opportunity to have developed this dissertation in such a welcoming environment.

To my mother, father, and older brother, I owe my deepest thanks for their unconditional support throughout my entire academic journey. To my girlfriend, thank you for always being there for me and for your constant encouragement and reassurance.

Finally, I would like to extend my appreciation to all the friends, colleagues, and teachers I had the privilege of working alongside during my academic journey.

Rodrigo Moucho

Declaração de Honra

Declaro que a presente dissertação é de minha autoria e não foi previamente apresentada e avaliada nesta ou em outra instituição. As referências a outros autores (afirmações, ideias, pensamentos), assim como a utilização de trabalhos publicados respeitam escrupulosamente as regras da atribuição, e encontram-se devidamente indicadas no texto e nas referências bibliográficas, de acordo com as normas de referenciação. Tenho consciência de que a prática de plágio ou de autoplágio constituem ilícitos académicos, conforme estabelecido no Código de Ética da U.Porto.

Declaro, ainda, que utilizei ferramentas de Inteligência Artificial para a melhoria da escrita da presente dissertação para aperfeiçoar a clareza, gramática e leitura da mesma. Estas ferramentas foram usadas exclusivamente para refinar texto já escrito e não contribuíram para as experiências e conclusões presentes nesta dissertação. Todas as sugestões geradas por ferramentas de Inteligência Artificial foram analisadas individualmente pelo autor e incorporadas quando apropriado. O autor aceita a responsabilidade total por todo o conteúdo presente nesta dissertação.

Rodrigo José Ramos Moucho



*“Perfection is achieved not when there is nothing more to add, but when there is nothing left to
take away”*

Antoine de Saint-Exupéry

Contents

1	Introduction	1
1.1	Context	1
1.2	Motivation	2
1.3	Problem	3
1.4	Research Questions	4
1.5	Publication	4
2	Literature Review	5
2.1	Literature Retrieval	5
2.1.1	Review Type	5
2.1.2	Methodology	5
2.1.3	Eligibility Criteria and Resulting Paper Set	6
2.2	Related Work	7
2.2.1	Background	7
2.2.2	Positional Bias in Convolutional Networks	7
2.2.3	Dataset Bias and Positional Effects	14
2.2.4	Positional Information in Transformer-Based Models	17
2.2.5	Summary and Outlook	18
3	Measuring, Analyzing, and Mitigating Translation Bias	25
3.1	Cork dataset and duplicate-defect setting	25
3.1.1	Dataset	25
3.1.2	Evaluation Pipeline	27
3.1.3	Model Training and Mitigation Strategies	28
3.1.4	Discussion	33
3.1.5	Influence of defect size on pair consistency	36
3.2	Periodic Confidence Modulation	38
3.2.1	Experimental Protocol	38
3.2.2	Autocorrelation Analysis	39
3.2.3	Existence of Structured Periodicity	40
3.2.4	Scale Covariance and Periodic Families	45
3.2.5	Architectural Origin: Dependence on Maximum Effective Stride	48
3.2.6	Cross-Dataset Validation and Periodic Confidence Structure	52
3.2.7	Interpretation and Implications for Positional Bias	56
3.3	Mitigating Confidence Variability	57
3.3.1	Experimental Protocol	57
3.3.2	Detection Robustness	58
3.3.3	Confidence Stability on Common Perfect-Recall Subsets	60

4	Conclusions and Future Work	63
4.1	Summary of Contributions	63
4.2	Revisiting the Research Questions	64
4.3	Implications	65
4.4	Limitations	65
4.5	Future Work	66
	References	67

List of Figures

1.1	Cork Stopper Setup	3
2.1	Common padding strategies and their associated boundary conditions. The red box illustrates the original image. Regions outside it are introduced by zero, reflect and circular padding, respectively.	8
2.2	Pooling-based downsampling. Source: https://epynn.net/Pooling.html	9
2.3	PASCAL and COCO dataset distributions excluding very large objects.	15
3.1	Left side of cork image shows a duplicated defect pair highlighted in pink. A singular (non-paired) defect is highlighted in green. The dotted red line shows when the cork rotates more than 360° and starts repeating visual content.	27
3.2	Two examples of the produced results after conducting the augmentation described in the Large-Scale Data Augmentation subsection. Each original image produces 3 augmentations. As exemplified in the top image, the pair defect structure is frequently destroyed. The right side of the cork stays untouched and is hidden in this Figure.	31
3.3	Two examples of the produced results after conducting the augmentation described in the Oversampling Pair Defect Images subsection. In contrast to the Large-Scale Data Augmentation pipeline, the pair defect structure is always preserved. The right side of the cork stays untouched and is hidden in this Figure.	32
3.4	Four images being detected using different training strategies.	35
3.5	Three examples of pair defects, with their bounding boxes highlighted in green, which belong to Q1	36
3.6	Three examples of pair defects, with their bounding boxes highlighted in green, which belong to Q4	36
3.7	Examples of 4 translations from the entire treadmill sweep. The left region of the image, including the defect (green box), is translated pixel by pixel for all valid positions.	39
3.8	Confidence sequence for the same image at five resolutions, with default inference $imgsz = 640$. The dominant period scales linearly with the image size.	41
3.9	Example of confidence variation and corresponding pixel shifts for a high-confidence detection. Notably, the confidence at pixel 32 returns to the exact value (0.946) observed at pixel 0.	42
3.10	Confidence trajectories for two datasets displaying large absolute confidence variations alongside the 32-pixel periodic structure. In both cases, the stride-induced periodicity is maintained.	43

3.11	Mean confidence versus confidence standard deviation across 171 perfect-recall 640×640 treadmill datasets. The negative trend (Pearson $r = -0.73$, Spearman $\rho = -0.80$) indicates that lower-confidence detections show greater variability. .	44
3.12	Confidence variation for an image in 640×640 resolution size inferred at <code>imgsz=1024</code> . Colors are used to improve the visualization of long patterns.	46
3.13	Confidence variation for image in 960×960 resolution size inferred at <code>imgsz=1024</code> . Colors used for improving visualization for long patterns.	47
3.14	Confidence variation for image in 960×960 resolution size inferred at <code>imgsz=1280</code> . Colors used for improving visualization for long patterns.	47
3.15	YOLOv8 architecture diagram provided by https://github.com/RangeKing	49
3.16	YAML configuration of the detection head for the default architecture (left) and the A1 ablation (right). The P5 detection branch and its connection to the Detect layer are removed, while the backbone remains unchanged.	50
3.17	YAML configuration of the default architecture (left) and the A2 ablation (right). The entire stride-32 backbone stage is removed together with the P5 detection branch, reducing the maximum effective stride from 32 to 16.	51
3.18	YAML configuration of the default YOLO backbone (left) and the A3 ablation (right). The only modification is the stride of the first convolutional layer, which is reduced from 2 to 1. This changes the maximum stride from 32 to 16.	52
3.19	Lag correlation plots for the same treadmill dataset across the three ablations. In A1, the 32 pixel periodicity remains intact. In A2 and A3, the dominant period shifts to 16 pixels.	52
3.20	Example images of the cross-dataset used for validation, defect boxes highlighted in green	53
3.21	Examples of the conducted zoom-in transformation. The top row includes images from the original dataset, and the bottom row displays their zoomed-in versions. .	54
3.22	Left: Per-head classification activation for example image: original model (top) vs zoom-in model (bottom). Detection shifts from P3 (stride 8) to P4 (stride 16). P5 is silent in both and is omitted. Right: corresponding autocorrelation curves shifting from lag 8 to lag 16 after retraining.	55
3.23	Left: Per-head classification activation for example image: original model (top) vs zoom-in model (bottom). P4 emerges from silence while P3 remains dominant, P5 is silent in both and is omitted. Right: corresponding autocorrelation curves shifting from lag 8 to lag 16 after retraining.	56
3.24	Detection Robustness results for tested configurations. Values reported as mean across three seeds. Seed-level standard deviations for complete failure, partial failure, and complete perfect recall are respectively: Baseline ($\pm 0.7\%$, $\pm 2.5\%$, $\pm 2.0\%$), BlurPool ($\pm 1.2\%$, $\pm 2.6\%$, $\pm 1.5\%$), Conv0Stride1 ($\pm 1.7\%$, $\pm 1.9\%$, $\pm 1.3\%$), Large-Scale Augs ($\pm 0.0\%$, $\pm 0.4\%$, $\pm 0.4\%$), Conv0Stride1 + Large-Scale Augs ($\pm 1.5\%$, $\pm 2.7\%$, $\pm 1.7\%$)	59

List of Tables

2.1	Summary of key literature on positional bias in computer vision	19
3.1	Pair Consistency Metrics Across Different Strategies	36
3.2	Duplicate defect consistency by defect scale.	37
3.3	Dataset image sizes and reported periodic pattern	41
3.4	Predicted Base Period P_0 from Equation 3.6 versus the dominant observed period P_{dom}	47
3.5	Confidence stability on common per-seed perfect-recall subsets (mean $\pm$ std across three seeds).	61

List of Acronyms

AP	Average Precision
AI	Artificial Intelligence
CNN	Convolutional Neural Networks
DETR	Detection Transformer
IoU	Intersection over Union
mAP	mean Average Precision
MS-COCO	Microsoft Common Objects in Context
PASCAL VOC	Pascal Visual Object Classes
SDG	Sustainable Development Goals
SSD	Single Shot MultiBox Detector
SOTA	State of the Art
ViT	Vision Transformers
YOLO	You Only Look Once

Chapter 1

Introduction

This introductory chapter outlines the topics that will be addressed in this dissertation. It presents the research context and motivation, formulates the problem to be solved, and sets forth the guiding research questions.

1.1 Context

Recent advances in the deep learning field, a branch of machine learning that utilizes neural networks, have led to great progress in a variety of areas, particularly computer vision research and development. Object detection is a core component in this domain and, by allowing the accurate localization and classification of target objects in images, has enabled an increasing list of practical applications for detection models. In a constantly evolving environment, convolutional neural networks (CNNs) and, more recently, vision transformer architectures, continuously set new benchmarks and define the state of the art (SOTA).

These advances have quickly increased the reliability of object detection models, making them useful to a wide range of real-world scenarios (e.g., autonomous driving [19]; manufacturing inspection [47]; medical imaging [29]; surveillance [22]).

Despite all the achieved progress, recent studies show that models suffer from systematic weaknesses. Notably, widely used traditional evaluation metrics, such as mean Average Precision (mAP) and accuracy, can oversimplify model performance by simply providing an aggregated score for the entire image, thereby ignoring potential large behavior variations for different zones inside the frame.

This limitation has been linked to a phenomenon known as positional (or spatial) bias [46], where detection capability systematically varies depending on the object’s location in the image, affecting even modern SOTA models. For instance, detectors often exhibit a marked performance drop for objects close to the image border [46], convolutional layers can encode and exploit absolute spatial location [24], and transformer-based classifiers, owing to their reliance on positional embeddings, can reinforce biases present in the training data [5].

1.2 Motivation

High performance on aggregated metrics may suggest a model possesses reliable detection. However, implicit positional bias may create serious risks, particularly in safety-critical applications, in which objects may appear anywhere in the image.

If a model consistently underperforms in certain regions, an object placed in an atypical location is much more likely to be misclassified or missed, with potentially catastrophic consequences.

For this reason, there is a strong motivation to study why and how positional bias arises in object detection systems, both for scientific purposes and practical applications. Understanding how this bias affects object detectors and moving beyond global metrics constitutes a pragmatic necessity for implementing more reliable models in real-world deployments.

Although it is an extremely relevant issue for a variety of settings, positional biases are particularly important in industrial environments, which are frequently tasked with detecting possible product defects and ensuring their quality.

In such environments, image acquisition is done in a controlled and consistent machine-based process, which excludes common external factors (e.g., differences in lighting or viewpoint) as a possible cause of detection performance variations.

A clear example is the analysis of cork stoppers for possible defects under a fixed setup as illustrated in Figure 1.1. Despite uniform image conditions, the same defect may have significantly different results when appearing in distinct positions, providing a concrete, yet generalizable case study for this research.

By identifying, quantifying, and reducing positional bias, object detection systems can become more spatially robust, reliable, and safe. To successfully address this issue, a clear understanding of its origins and manifestations is required, so that the strategies to measure and mitigate its effects can be correctly developed.

Throughout this thesis, the term positional bias is used as a general expression for variations in detection performance due to an object’s location. When this dependency is analyzed through direct translation of an object’s position under consistent appearance and scale, we broadly refer to it as translation bias.

While the literature review explores positional bias more generally, the experimental work will specifically concentrate on translation bias. Furthermore, our analysis will focus exclusively on 2D models and center on the **CNN** architecture.

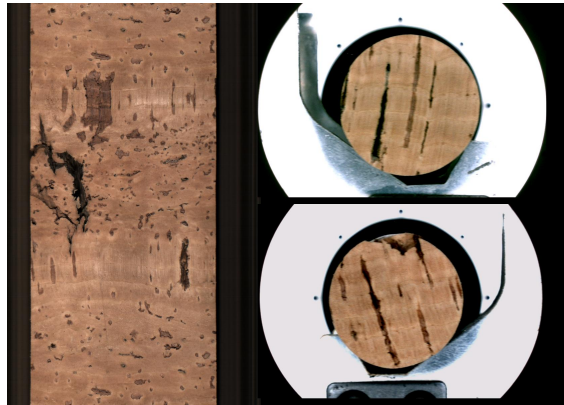


Figure 1.1: Cork Stopper Setup

1.3 Problem

In recent years, wider research on positional bias has led to a better understanding of this issue and increasing recognition that **SOTA** object detection suffers from it. Still, it remains insufficiently characterized and underexplored.

- Evaluation gap: Metrics such as **mAP** remain the dominant standard, completely ignoring spatially localized weaknesses. Evaluation methods that take into account translation sensitivity are not yet a usual practice.
- Understanding gap: While **CNNs** and Transformers have been shown to encode positional information, it is still discussed how strongly different architectures and datasets contribute to bias under varying spatial distributions.
- Mitigation gap: Although several techniques have been proposed to reduce positional bias, their practical effectiveness and usability still vary for different applications and scenarios. Additional research is warranted to explore and validate other strategies in specific application domains.

Tackling these gaps is required to improve the comprehensibility of positional bias and enhance the reliability of modern detection systems, making it an extremely relevant problem:

- For the research community, by contributing to the existing literature and highlighting limitations of traditional metrics.
- For industry and safety-critical applications, where the existence of positional bias has highly relevant risks.
- For future deployments, by exploring mitigation strategies, particularly for translation-induced positional bias, leading to more trustworthy detectors and increased applicability in real-world contexts.

1.4 Research Questions

- RQ1. To what extent do modern object detection models exhibit translation-induced positional bias, and how does this bias manifest in a controlled industrial setting?
- RQ2. What architectural and training-related factors govern translation bias in CNN-based detectors, and how can these mechanisms be quantified and visualized?
- RQ3. What training strategies, data augmentations, or architectural modifications can mitigate translation bias, and how do these strategies compare with each other in terms of effectiveness, trade-offs, and complementarity?

1.5 Publication

Part of the work presented in this dissertation has been submitted to the British Machine Vision Conference (BMVC) 2026.

Chapter 2

Literature Review

2.1 Literature Retrieval

The following section contains the methods employed for the search, collection, screening, and selection of relevant papers to be analyzed and researched. The objective is to define and describe the literature review process employed to identify prior work addressing positional bias and related object detection issues.

2.1.1 Review Type

The literature review follows a semi-systematic review approach, combining structured search with exploratory analysis. Since positional bias is a very extensive topic with various components, the review balanced reproducibility with methodological flexibility.

In the early stages of the review, an exploratory approach was used, combining seed papers recommended by the external supervisor, snowballing, and general queries filtered for relevance. The process prioritized broad topical coverage. This included diverse architectures (CNNs, transformer-based), their development, and their relation to positional bias, as well as different types of data biases and practical implications. Snowballing and AI-based retrieval tools complemented traditional database searches.

2.1.2 Methodology

The review process was iterative, integrating query refinement while expanding the search through reference-based exploration. To begin the literature review, the recommended seed papers were thoroughly reviewed. As mentioned, throughout the entire process, a large number of additional papers were retrieved by using snowballing and analyzing the references of important works.

General search queries were run on Google Scholar and sorted by relevance. These include:

- Q1. spatial bias object detection
- Q2. convolutional neural network "translation invariance" positional bias

- Q3. vision transformer "translation invariance" positional bias
- Q4. vision transformer positional location bias
- Q5. visual dataset bias computer vision

Synonyms such as "spatial" and "positional" were alternated during searches to broaden coverage.

During paper reviews, specific knowledge gaps were found that could not be adequately addressed through reference-based search. Therefore, an additional query was created for greater understanding of padding-related bias:

- Q6. CNN padding spatial bias

Lastly, AI-powered semantic search tools were also employed using natural language prompts that matched the above keywords. These were particularly useful to find important papers for which conventional database query searches were not returning effective results.

2.1.3 Eligibility Criteria and Resulting Paper Set

In order to filter out inapplicable or irrelevant papers, the following strategy was employed:

- Papers must explicitly mention topics relevant to this dissertation, preferably in the title or the abstract.
- The focus should be on object detection in computer vision, although classification or segmentation was also included if they contained pertinent information.
- Publications were required to be written in English, and papers should be published in reputable journals after 2015, although exceptions may occur. These should preferably have a high citation count. However, less-cited papers were also included when they were recent or presented particularly compelling and strong findings.

Conversely, unless they exhibited transferable insights, works that discuss non-relevant bias were discarded, as were papers that focused on Natural Language Processing, Vision-Language Models or 3D Vision Models.

The resulting paper set consists of 27 research papers and constitutes the core literature used to describe and understand positional bias. These works span spatial bias in computer vision, dataset and contextual bias, modern object detection architectures, and evaluation or mitigation strategies relevant to this dissertation. In addition to this set, further works are included where relevant, even if they are not directly related to positional bias.

2.2 Related Work

The following section reviews the literature on positional bias in modern vision detectors. It includes a concise overview of the evolution of object detection architectures, examines architectural and dataset-level mechanisms that induce spatial bias, and evaluates existing approaches for quantifying and mitigating these effects.

Although the experimental component of this work is exclusively related to object detectors, the majority of existing literature on positional bias is centered on classification tasks. Nevertheless, underlying mechanisms are generally task-agnostic and findings transferable between tasks, thereby providing relevant insights.

2.2.1 Background

Object detection models have rapidly evolved throughout the last few years and continue to do so, making computer vision a popular and continuously improving field. This process of evolution involves constant architectural innovations driven by the need to address specific problems or inefficiencies that exist in detectors.

For instance, one-stage detectors, such as You Only Look Once (YOLO) [35] and Single Shot MultiBox Detector (SSD) [28], were developed to decrease computational costs, enabling real-time predictions. Meanwhile, many models incorporated multi-scale feature pyramid networks to enhance detection for objects with varying sizes, particularly smaller ones [2, 10].

Despite the advancements achieved by object detection models in recent years, researchers note that modern detectors, regardless of architecture design, have been shown to suffer from spatial biases [46]. Such an important issue requires further exploration, understanding, and applicable mitigation strategies, particularly for object detection contexts, which have not been given as much attention as classification tasks have.

2.2.2 Positional Bias in Convolutional Networks

2.2.2.1 Padding and Boundary Effects

Padding is a mechanism that is utilized in CNNs to control input spatial dimensions and handle image boundary effects. By adding artificial values to the borders of an image or feature map, padding increases the number of kernel placements during convolutional operations that include border pixels, thereby reducing the systematic under-representation that would occur if no padding were employed.

There is a large variety of padding strategies, resulting in different boundary conditions. Zero padding surrounds the image with zero-valued pixels, creating sharp discontinuities at image borders, while reflect padding mirrors image content to reduce edge artifacts. Circular padding makes the image wrap around its borders, being useful for periodic signals, but often not for natural images (Figure 2.1).

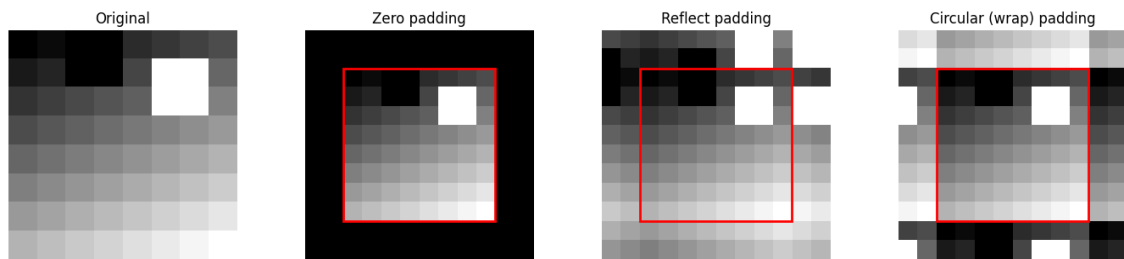


Figure 2.1: Common padding strategies and their associated boundary conditions. The red box illustrates the original image. Regions outside it are introduced by zero, reflect and circular padding, respectively.

The most common padding type is zero-padding, and although it is a fundamental component in CNNs, research indicates that its use is a source of positional bias.

As one study demonstrates [1], CNNs can develop "blind spots" by applying padding unevenly. This creates detectable artifacts that can propagate throughout the whole network, which in some cases can even produce certain regions where small objects are virtually undetectable. Additionally, many padding schemes are tested and compared in order to mitigate these artifacts, as well as to underline the different trade-offs each strategy entails, such as longer training and inference times for more complex methods. Symmetrical and circular padding are two proposed alternatives that are able to return compelling results by diminishing padding-induced effects.

Expanding on this idea, another approach [33] employs an original method named Peripheral Prediction Padding (PP-Pad) that learns padding values dynamically. Rather than simply utilizing a traditional padding method, PP-Pad uses convolutional layers to learn the optimal padding values for a specific image. When tested for translation invariance, the new strategy was able to improve it without decreasing overall performance, even though it significantly increased computational costs.

Another analysis further explores how zero-padding can be a source of absolute positional information encoding. They show that removing padding dramatically decreases a network's ability to infer an object's absolute location. Their findings confirm that different padding strategies lead to very distinct scenarios and validate that this mechanism impacts not only the borders of an input image but can also affect its center positions [18].

Additionally, in CNN architectures with a bigger receptive field, these effects are exacerbated. It has been demonstrated how a single convolutional layer, followed by global max pooling, is able to classify absolute position locations, contradicting the notion of translation invariance [24]. In order to remove the network's ability to encode position, researchers introduced an approach where convolutional filters are applied uniformly at all spatial locations, termed full convolution. This modification yielded better results in terms of stability in image shifts, providing further evidence that CNNs exploit border conditions.

The mentioned padding-induced artifacts can contribute to performance disparities in object detection tasks. Many studies find that standard CNN detectors consistently perform worse on

objects near the borders of an image [46] [41], particularly when trained on datasets with centered-skewed data, and argue this effect can also be partly traced back to border handling. In [41], toroidal circular padding is also experimented with, greatly improving performance on objects placed in the border.

These findings underscore the importance of carefully handling padding and borders in CNNs, and how they can unintentionally introduce positional bias and influence overall performance in modern detectors.

2.2.2.2 Stride, Pooling, and Shift-Equivariance

The vast majority of modern CNNs incorporate downsampling operations in their architectures (Figure 2.2). These modules reduce the spatial dimensions of feature maps, decreasing computational costs and enabling deeper architectures, making them critical components that CNNs depend on to gradually learn object representations. However, these operations (such as strided convolutions or pooling layers) have been found to break translation equivariance.

A model is said to be shift-equivariant if shifting the input by a given spatial translation T_Δ results in an equivalent shift in the output, without altering its structure or values:

$$f(T_\Delta x) = T_\Delta f(x) \quad (2.1)$$

A model is said to be shift-invariant if spatially shifting the input does not change the output:

$$f(T_\Delta x) = f(x) \quad (2.2)$$

In the context of object detection, these two concepts apply to different components. Spatial outputs such as predicted bounding box locations are expected to be equivariant. Values such as object class labels, confidence scores, and number of detections (for identical scenes) should remain invariant to translations.

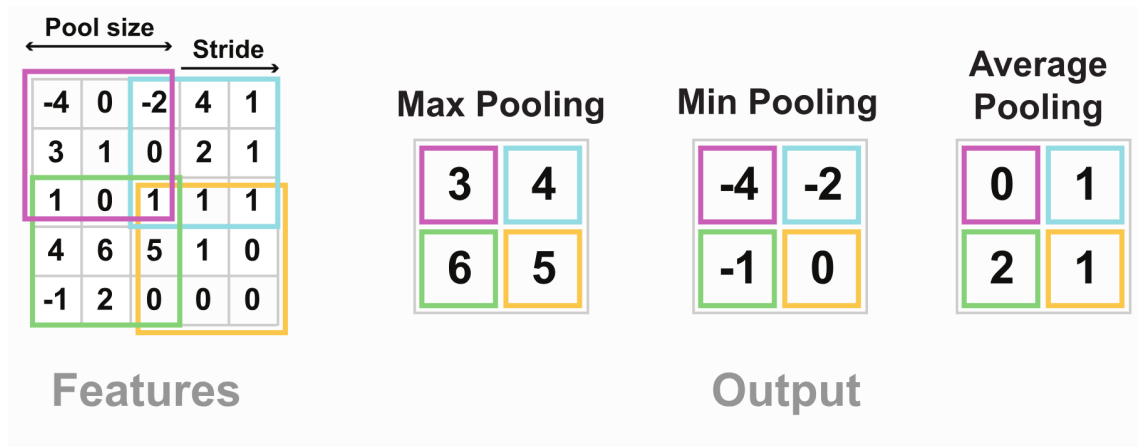


Figure 2.2: Pooling-based downsampling. Source: <https://epynn.net/Pooling.html>

Recent literature indicates that the introduction of subsampling in CNNs creates an unintended consequence: small translations can cause significant changes in the output due to how feature activations alter their alignment with the sampling grid. This happens because if the shift is smaller than the stride, previously sampled features may become unsampled, possibly leading to loss of valuable information, which propagates throughout the network.

Prior work demonstrates precisely this [3], showing that negligible pixel shifts can greatly alter a network’s predictions. These researchers attribute this to subsampling, more specifically to the violation of the sampling theorem.

The sampling theorem states that a signal can only be subsampled without loss of information if it does not have spatial frequencies (how quickly values change across space) higher than half the sampling rate. In practice, this means that when feature maps contain high-frequency spatial components, such as sharp activations, aliasing might be introduced during sampling operations, which in turn can make representations sensitive to small shifts. These small differences, compounded throughout the network, can lead to significant changes in the final prediction.

Subsequent research has attempted to solve or at least mitigate this issue by using a well-established technique: blur filters. Although faulty implementations may degrade overall performance, when correctly integrated, applying layers like BlurPool [44] before the downsampling operations smooths high-frequency features, thereby avoiding aliasing and achieving better shift-invariance with an uncomplicated approach.

The trade-offs between accuracy, shift-invariance, and complexity must always be considered when proposing architectural strategies. Since models can sometimes even benefit from biases, successfully implementing architectural solutions that improve a model’s generalization capacity can often lead to decreases in overall accuracy, as well as increased computational costs [31]. These are some of the reasons why many solutions are still not integrated into widely used object detection models, where efficiency and overall performance are often the priority.

A variety of architectural mitigation strategies have been developed in recent years. One approach developed a subsampling scheme named adaptive polyphase sampling (APS) to replace regular downsampling. This non-linear layer selects which image components to use, applying the same criterion to shifted images and retrieving consistent feature maps from both. Using this strategy, they were able to achieve near-perfect shift-invariance [6]. Similarly, a new pooling operator named Translation Invariant Polyphase Sampling (TIPS) introduces a mechanism to obtain more consistent outputs under shifts, successfully attaining better shift-invariance [37]. Together, these works serve as a powerful proof-of-concept for how architectural innovations can replace traditional methods and help solve this specific problem.

Another study [32] that thoroughly explored stride, reported that if there is high local homogeneity in feature maps (very similar neighboring pixels), subsampling can actually be beneficial to invariance. On the contrary, if there are high-frequency feature differences, invariance worsens, confirming previous findings.

The authors also argued that networks can conserve perfect equivariance for translations of multiples of the stride. Their analysis emphasizes that different datasets and architectures can be

distinctly affected by this phenomenon.

Dataset augmentation techniques are frequently mentioned as useful for positional information mitigation, alongside tweaks in loss functions [17]. Some works give a stronger importance to dataset quality and defend that better invariance results can be achieved using spatially augmented datasets rather than architectural changes [4]. Through enriching datasets by placing objects in multiple locations and avoiding unseen positions for individual classes, models can become more invariant to translation.

To sum up, the widespread use of downsampling operations in CNNs is a major source of positional and translation bias. As evidenced in the current chapter, these components break the translation equivariance for object translations and can create significant changes to predictions based only on the location of the input. Promising solutions to mitigate this problem are already being developed and showing favorable results, though their incorporation into SOTA models remains limited.

2.2.2.3 Novel Modules for Invariance

In order to enforce spatial invariance in computer vision detectors, researchers have also experimented with original methods that go beyond padding and pooling variations.

One of these innovations [38] seeks to allow models to focus on objects' representations rather than their locations, through architectural changes. To do this, the authors develop Patch Reordering, a strategy that ranks feature map patches and combines them with CNN operations. This results in object representations becoming invariant to position because that criterion yields consistent results, regardless of an object's exact position. The authors show this module improves benchmark classification tasks, and constitutes an intriguing approach, proposed before ViTs, that shares many conceptual similarities with later patch-based architectures.

A related work aims to achieve general equivariance in CNNs for translation, rotation, and scaling [40]. In order to accomplish it, they create a Similarity Convolution layer, which is a method that essentially becomes aware of transformations done to an image and uses that information to adapt the employed convolutional filters, demonstrating robust results. This proposal goes beyond pure positional bias but serves as an example of how ambitious and innovative design is an important path to improving generalization in vision models.

It should be noted that these modules frequently increase integration complexity or computational costs, and many focus on specific classification tasks, which may not be representative of general-use cases or necessarily translate to object detection. Nonetheless, they demonstrate potential routes that fundamentally alter how CNNs function and produce valuable experiments and ideas [40].

2.2.2.4 Traditional Evaluation Metrics

Object detection models are generally evaluated using a small number of predefined metrics to quantify detection quality. These metrics are used as a baseline for evaluation and are paramount

for objective comparisons between model performance on specific datasets.

- Intersection over Union (IoU)

IoU measures the geometric overlap between a predicted bounding box and the corresponding ground-truth box. Given a predicted bounding box B_p and a ground-truth bounding box B_{gt} , IoU is defined as:

$$\text{IoU}(B_p, B_{gt}) = \frac{|B_p \cap B_{gt}|}{|B_p \cup B_{gt}|} \quad (2.3)$$

- Precision and Recall

Precision measures the proportion of correct predictions out of all made predictions, while recall measures the proportion of objects which are successfully detected. These two metrics complement each other and can be defined as:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \quad \text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \quad (2.4)$$

where TP, FP, and FN denote the number of true positives, false positives and false negatives, respectively. Recall is a fundamental metric in this study because it allows for quantifying the percentage of missed objects, being particularly relevant for analyzing positional effects.

- Average Precision and Mean Average Precision

Average precision is measured by calculating the area under the Precision-Recall curve and is defined as:

$$\text{AP} = \int_0^1 p(r) dr \quad (2.5)$$

where $p(r)$ denotes the precision at recall level r

Mean Average Precision (**mAP**), which is more commonly used, is calculated by using the average of **AP** across all dataset classes.

$$\text{mAP} = \frac{1}{C} \sum_{c=1}^C \text{AP}_c \quad (2.6)$$

where C denotes the number of object classes in a specific task. **mAP** is frequently measured at different **IoU** thresholds to evaluate detectors with different prediction precision conditions.

As previously mentioned, although these metrics provide a widely accepted and consensual evaluation framework, they solely provide a global score. This results in a complete lack of quantification as to how results differ between different image locations, hiding possible spatial flaws

the detectors may possess. This limitation motivates the development of location-level evaluation techniques, which will be explored during this dissertation.

2.2.2.5 Measuring and Visualizing Spatial Sensitivity

Evaluation Frameworks for Spatial Sensitivity In recent years, several researchers have begun to develop evaluation frameworks to measure the positional effects that are not represented in traditional detection metrics. Such frameworks often focus on quantifying different components of spatial and shift biases, whether through controlled input transformations or image partitioning to isolate position as the crucial variable.

Studying shift equivariance, a major architectural cause of translation bias, existing work [30] introduces a methodology to measure end-to-end shift equivariance by evaluating shifted versions of the same image embedded in a larger black background. Their framework analyzes object detection consistency under numerous small global image translations and explores strategies to improve it. The conducted experiments exemplify the importance of having a valid and effective evaluation tool for investigating this topic.

A complementary study follows a different approach [46] and analyzes how **SOTA** models function spatially by utilizing a zone-based protocol. This method splits the image into predefined spatial regions (annular zones) from the center to the borders. By analyzing zones independently, this technique enables a direct comparison of detection quality across image locations and exposes systematic positional variations, unmasking global performance measures and demonstrating the importance of explicit zone analysis.

Currently, there is no unified testing framework for positional bias in object detection. This is partly due to the fact that this phenomenon can be analyzed from numerous perspectives; some researchers focus on discrepancies after image translations, while others focus on exploring how detection performance evolves across spatial regions. Nevertheless, the existing frameworks demonstrate a growing recognition of positional bias as a fundamental phenomenon in computer vision and provide methodological foundations for its analysis.

Empirical Analysis and Model Comparisons As a way to quantify the consequences of positional bias in an objective, interpretable way, recent works have proposed methods and conducted extensive experiments to systematically measure this phenomenon and compare it between architectures.

Some initial studies conducted on this topic, such as [23], argue that data augmentation is the most important factor in obtaining translation invariance. The testing of many **CNN** architectures trained on augmented and non-augmented data, as well as testing deeper networks and design choices, allows for direct comparison between design choices, helping to establish the generality of the problem and shape future investigations.

In a more recent study, one approach tests modern **CNNs** on millions of images with objects at various locations of the image frame and measures both global and local shift invariance. They found patterns such as a **CNN** being stable for shifts in certain directions or areas but displaying

high instability in others, many without any clear explanation why, highlighting uneven spatial behavior [26].

Overall, the existing literature provides rigorous research on architecture choices and how they influence positional bias, alongside structured visualization and model-comparative studies that confirm it as a tangible, widespread problem that must be addressed and understood more deeply.

2.2.3 Dataset Bias and Positional Effects

Robustness is one of the central evaluation axes for computer vision models, expanding beyond accuracy to assess how reliable a detector is under distributional and spatial shifts. Empirical evidence shows that increasing model size and training data volume leads to disproportionate improvements in model robustness, particularly for out-of-distribution conditions, involving variations in object location [11]. These findings indicate that positional bias, particularly in the form of translation bias, is not exclusively induced by architecture design, motivating a focused analysis of dataset-induced effects.

Large amounts of visual data are essential for successfully training modern computer vision models. However, many datasets suffer from systematic biases that can skew network performance, including selection bias, label bias, or framing bias [15]. Learning how to identify these biases and understanding how consequential they can be to model behavior are key steps towards the design of robust and reliable computer vision systems.

Dataset bias can be defined as the systematic imbalances a dataset possesses that emerge from the way data is collected, annotated, or distributed, and that create learnable patterns that AI models can exploit. In object detection, these biases can manifest, for example, as non-uniform distributions of object placements.

2.2.3.1 Datasets

A vast array of vision datasets is publicly available for developers to utilize for different tasks. The research community has adopted a select set of them as benchmarking datasets for fair and uniform comparison and evaluation between model architectures.

- **PASCAL VOC** [14]:

The PASCAL Visual Object Classes (**PASCAL VOC**) dataset is one of the earliest benchmarks for classification and object detection. It contains 20 classes, including common objects such as person, car, and dog. The most recent version (VOC12) has about 11000 images, which is considerably less than newer large-scale datasets. Nowadays, due to its smaller size, **PASCAL VOC** is more often used for controlled experimentation or quick prototyping and evaluation.

- **MS-COCO** [27]:

The Microsoft Common Objects in Context (**MS-COCO**) is one of the definitive benchmark datasets and reflects complex real-world scenes, being primarily known for object detection and segmentation tasks. It contains over 330000 images and includes 80 general object categories, making it a challenging, large-scale benchmark for detection research.

- ImageNet [8]:

ImageNet is another large-scale database widely used for image classification and pretraining backbones for a variety of other tasks. A very commonly used subset is ImageNet-1k, which contains 1000 classes and over 1.2 million training images, making it one of the largest available datasets. The complete dataset has over 14 million images and has been a crucial resource for computer vision advancements.

- CIFAR-10 [25]:

CIFAR-10 is a small-scale image classification dataset consisting of 60000 images, divided into 10 classes, such as airplanes, dogs, and cars. This dataset is frequently used for methodological studies, such as the analysis of model behavior under image transformations.

For this dissertation, experiments will be mostly conducted using an industrial cork defect dataset due to its favorable characteristics for studying this phenomenon.

In object detection, **MS-COCO** 2017 and **PASCAL VOC** 2012 are frequently used as representative benchmarks due to their widespread adoption. Their object distribution is shown in Figure 2.3, where each object is placed according to its center, revealing a clear distribution concentration towards the center of the image. Very large objects (area > 50% of image) that unfairly skew the data to the center of the image are not included to ensure a more informative representation.

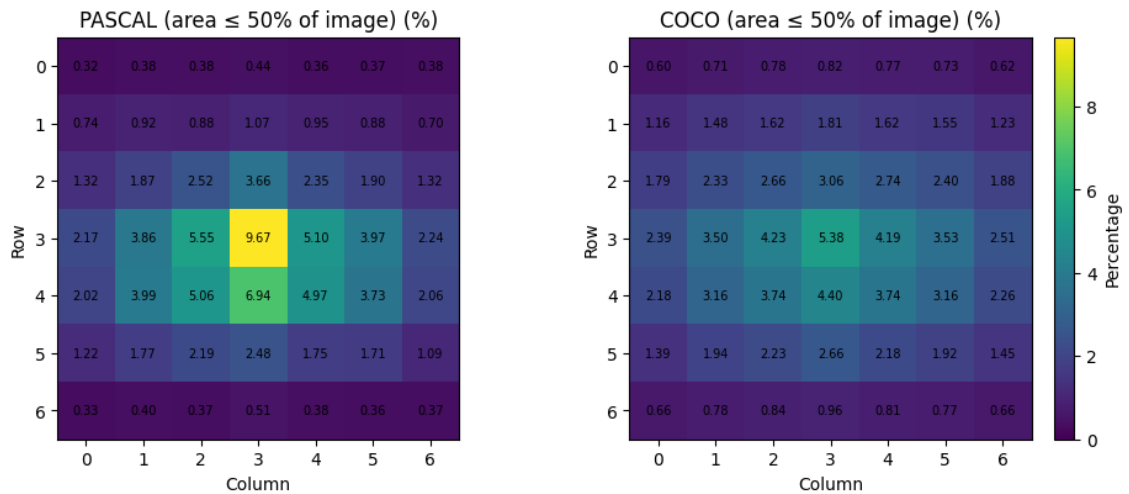


Figure 2.3: PASCAL and COCO dataset distributions excluding very large objects.

2.2.3.2 Position Bias in Training Data

A prominent bias that datasets suffer from is photographers' bias, also referred to as center bias, where objects appear overwhelmingly in the center of the image frame to the detriment of the borders. This issue arises due to the tendency humans have to center the subject of an image, as well as from data collection biases, and it can be a significant source of positional bias [3, 30, 46].

Since learning-based models are inherently dependent on their training data, models trained on such datasets can implicitly learn these biases. Specifically, they learn to expect objects of interest to be located in the center of the image frame, which raises issues for objects placed in out-of-distribution locations.

This problem is further exacerbated by the fact that computer vision models are not truly shift-invariant. Several studies trained modern models on widely used datasets (such as MS-COCO) and have reported significant drops in performance for images with objects near the border [41, 46]. This problem affects all types of architecture families and is especially concerning for safety-critical tasks where objects may appear anywhere in the image.

Data-centric mitigation strategies focus on refined data augmentation processes, such as increasing the sampling of objects in low-frequency zones, which yields respectable improvements. Other strategies include zone-aware loss functions that prioritize border objects, making them receive more attention, as well as modifications to padding operations. While these approaches can minimize zone variance, they may introduce relevant trade-offs, such as overall performance degradation.

Related work has also studied this issue in ViTs. One study measured the level of position bias in a number of datasets with varying levels of spatial bias (Position-SHAP) and evaluated model behaviour [5]. By enabling the model to adapt its positional sensitivity to match the bias present in the training data, the authors show, as expected, that datasets with strong positional biases benefit from higher Position-SHAP, while the contrary happens for datasets with lower bias. These findings underscore how crucial it is to account for the biases present in datasets.

2.2.3.3 Other Dataset Biases

As previously mentioned, understanding the biases inherent in datasets is of paramount importance. For this reason, examining the various facets of dataset bias is beneficial for a broader comprehension of model behavior, even when such biases are not directly related to positional issues.

Another well-documented bias is context bias. This bias manifests itself when models use objects that frequently co-occur or background cues as shortcuts to improve accuracy. Instead of focusing on the object itself, the model relies on contextual information surrounding the object, making this a complementary problem to positional bias.

As an explanatory example, a model trained on images of the category "skateboard" where it consistently co-occurs with "person" can lead to classifier over-reliance on this association, which does not always exist. This context bias can lead to poorer results when a skateboard appears

by itself, out of its usual context [39]. Recent explainability research also explored this issue and reached similar conclusions, confirming and detailing how detection models use background concepts and co-occurring categories to improve predictions [13].

In addition to enriching training data, a logical mitigation strategy is to penalize the model when it overly relies on these context cues instead of the actual object. These efforts seek more robust and unbiased networks, but must carefully balance their approach since exploiting an object’s context can frequently be advantageous to the detector.

Another long-recognized fundamental problem is the intrinsic bias in datasets. This originates from different data collection and annotation techniques and can cause a lack of cross-dataset generalization, meaning a model can achieve strong results for in-dataset images but produce much poorer results for the same categories in samples from different datasets.

Pioneer work in [42] drew attention to this issue and measured how biased well-known datasets were. They did this by training early object detectors on a dataset and testing on different ones, creating a cross-dataset generalization table that confirmed the severity of the problem at hand.

More recent research keeps confronting this issue by developing tools to mix multiple datasets with their inherent biases in order to improve the model’s robustness in real-life practical applications [34].

Although these works do not directly address positional bias, they still offer valuable insights into how systematic data biases can deeply influence model behavior.

2.2.4 Positional Information in Transformer-Based Models

Transformer architectures, such as ViTs and Detection Transformer (DETR)-like detectors, handle positional information differently from CNNs. By decomposing the image into a set of patch tokens and feeding them to the model as a sequence, the self-attention mechanism is agnostic to each token’s position unless it is explicitly provided [12].

Positional Encodings (PE) are a crucial component in transformers by injecting location information (via sinusoidal patterns, learned embeddings, or relative position), enabling detectors to understand spatial relationships. The explicit handling of this info frequently creates benefits, but it can also produce weaknesses.

On one hand, a ViT has fundamentally different mechanisms from CNNs. In the absence of PEs, it is entirely agnostic to absolute position, which can promote certain forms of invariance but raises challenges for tasks that can leverage positional information. On the other hand, transformers are not shift-invariant, and, similarly to CNNs, can make inconsistent predictions even for small translations.

Recent research shows examples of this lack of translation equivariance. For instance, sliding an image by a few pixels may alter a classifier’s accuracy by as much as 20 percentage points, which can be traced to patch embedding, positional encoding, and subsampled attention mechanisms [9].

Mirroring approaches followed in CNNs, recent works have proposed redesigned components that directly address this issue, although mostly focusing on small image shifts.

The authors of [36] introduce a variety of adaptive modules for tokenization, window-based attention, patch merging, and positional encoding that achieve circular shift invariance by aligning operations in a data-dependent manner and ensuring consistent outputs under shifts. Likewise, in [9], shift-equivariance algorithms are developed and integrated with ViT models, successfully attaining input shift consistency. Notably, the work does not target state-of-the-art accuracy, but instead focuses on controlled academic experiments, emphasizing the existing challenges of integrating these types of modules into industrial-scale systems.

As mentioned in the Dataset Bias section, recent work [5] shows that, depending on the task, models can have positive or negative consequences from the positional information they learn from datasets. Their experiments highlight the importance of positional embeddings and how their significance varies greatly depending on the bias present in the data.

In complementary but still relevant work, Relation-DETR [16] provides a novel approach that captures the positional correlation between objects in real datasets and injects it into the attention layers. By explicitly utilizing the knowledge of common object positions and their co-occurrences, the authors are able to benefit from biases, illustrating how the main goal of researchers should be to avoid unwarranted or harmful biases and not to discard them altogether.

Adding to this, prior studies [43] have conducted a systematic review of relative position encoding methods and demonstrate that relative positional encoding can fully replace absolute positional encoding for classification tasks, but not for object detection. While classification performance improves (or remains stable) using relative encodings, detection performance deteriorates when absolute position information is removed.

This study exposes how positional encoding mechanisms are intrinsically task-dependent and how conclusions drawn from classification tasks do not always transfer to object detection. Such evidence is particularly relevant for this dissertation since most of the existing literature on positional and translation bias centers on classification settings, while the experimental work will focus on object detection.

The collective efforts of the computer vision research community are gradually increasing the understanding of how position affects transformers, through innovations in adaptive modules or positional embedding designs. The emergence of next-generation detectors that will allow models to generalize across the image frame and recognize objects regardless of location is an attainable, yet challenging, goal.

2.2.5 Summary and Outlook

Research in recent years has increasingly shown that positional bias is a wide-ranging, multi-faceted topic that significantly affects object detection systems.

These effects are not particular to a single architectural design. Padding and subsampling break shift-equivariance, which can cause translation inconsistencies, while transformers exhibit similar issues due to both architectural design and the use of positional embeddings. The existing literature diagnoses these problems through theoretical analysis and exhaustive practical and visualization

experiments, and has begun to develop a variety of possible solutions to reduce or control the effects caused by this phenomenon.

The mitigation paths researchers follow fall under two major categories:

- Dataset-focused approaches, which mainly include data augmentation or balancing techniques
- Architectural innovations, which include anti-aliased CNN layers, padding strategies and adaptive transformer modules

This collection of discussed papers allows for a thorough review of the current literature, some of which focus on a deep foundational understanding of positional bias; others propose practical fixes and model components that directly address shift variance, and others shine light into related topics that have rich transferable insights.

The lessons taken from these works indicate that combining complementary approaches is the optimal strategy for achieving successful outcomes. Ultimately, they form a solid basis for the thesis to build upon and confirm the importance of fairer, more robust, and more generalizable detection models.

In table 2.1, a summary of the works that analyze, mitigate, or explore model positional information is included, serving as a condensed overview of the most relevant literature on the topic.

Table 2.1: Summary of key literature on positional bias in computer vision

Reference	Year	Task / Model Scope	Main Contribution
[1]	2020	Object Detection (CNN)	Demonstrates that zero padding can induce spatial artifacts and blind spots in CNN feature maps, leading to severe detection failures near image boundaries. Proposes padding-related mitigation strategies.
[3]	2019	Image Classification (CNN)	Quantifies severe translation sensitivity in CNNs and explains it via subsampling-induced aliasing and dataset bias. Shows that convolution and data augmentation do not guarantee invariance, and that existing mitigations are partial.

Reference	Year	Task / Model Scope	Main Contribution
[5]	2025	Image Classification (ViT)	Introduces Position-SHAP to quantify positional bias in datasets, showing that the effectiveness of positional encodings depends on dataset bias. Proposes an adaptive positional encoding mechanism to match sensitivity.
[6]	2021	Image Classification (CNN)	Demonstrates that downsampling (stride) is the primary cause of shift variance in CNNs. Proposes Adaptive Polyphase Sampling, a non-linear subsampling scheme that achieves perfect shift-invariance without accuracy loss, outperforming data augmentation and anti-aliasing methods.
[11]	2021	Image Classification (CNN)	Studies out-of-distribution robustness and transferability in modern CNN s. Shows increasing training data, model capacity, and preprocessing changes improve robustness. Introduces a synthetic dataset for robustness evaluation in a controlled environment.
[13]	2022	Object Detection & Segmentation (CNN)	Proposes a concept-based explanation method, exposing model reliance on background and contextual features, enabling systematic detection of context bias.
[15]	2022	Visual Datasets (Survey)	Provides a survey of bias in visual data (selection, framing, and label), reviews methods for bias discovery, and analyzes bias-aware construction practices.
[16]	2024	Object Detection (DETR)	Introduces explicit position relation priors by encoding the correlation between objects. Shows how this strategy can improve detection accuracy and accelerate convergence.

Reference	Year	Task / Model Scope	Main Contribution
[18]	2021	Image Classification & Segmentation (CNN)	Provides analysis showing zero padding and border handling inject absolute position information. This drives spatial bias, affects semantic learning, and can be beneficial or detrimental depending on the task.
[24]	2020	Image Classification & Matching (CNN)	Shows that standard convolutional layers can encode spatial position via boundary effects, breaking translation invariance. Proposed Full Convolution to remove location bias, improving robustness and data efficiency.
[26]	2020	Feature Extraction / Image Classification (CNN)	Provides a quantitative study of shift invariance in off-the-shelf CNN features, showing a lack of shift-invariance and strong positional bias.
[32]	2021	Image Classification (CNN)	Provides analysis of stride in CNNs, showing subsampling can improve translation invariance when combined with high local homogeneity. Analyses and compares methods and respective trade-offs
[33]	2023	Semantic Segmentation (CNN)	Proposes Peripheral Prediction padding, which enables task-specific training of padding values, reducing positional encoding induced by zero padding. Introduces quantifying metrics for translation invariance and shows improved results.
[36]	2023	Image Classification & Segmentation (ViT)	Identifies that standard ViT are not shift-invariant due to fixed tokenization, window-based self-attention, patch merging, and positional encoding. Proposes adaptive, data-dependent module redesigns achieving shift consistency.

Reference	Year	Task / Model Scope	Main Contribution
[37]	2024	Image Classification, Object Detection & Segmentation (CNN)	Introduces Translation Invariant Polyphase Sampling, a pooling operator that reduces maximum-sampling bias in downsampling layers, demonstrating an inverse relationship between pooling bias and shift invariance. Achieves improvements in shift consistency and overall performance across different tasks.
[38]	2019	Image Classification & Retrieval (CNN)	Proposes Patch Reordering, an architectural module that reorders feature-map patches to encode translation and rotation invariance. Produces consistent representation under global spatial transformations and improves robustness.
[41]	2023	Dataset Bias & Positional Bias (CNN)	Demonstrates that common vision datasets exhibit strong center bias, which interacts with boundary conditions to cause performance degradation near image borders. Shows data shifting and toroidal boundary conditions help mitigate boundary bias
[44]	2019	Image Classification (CNN)	Demonstrates that lack of shift-invariance in CNNs is primarily caused by aliasing introduced by strided and pooling operations. Proposes anti-aliasing before subsampling, improving translation robustness and accuracy.
[46]	2024	Object Detection (CNN, DETR)	Introduces Zone Evaluation, a spatial evaluation protocol that measures detection performance across zones to detect spatial bias. Demonstrates that detectors consistently underperform near image borders, which is masked by the overall metrics. Shows that bias is caused by zone-specific data patterns and proposes spatially balanced learning strategies to reduce zone variance.

Reference	Year	Task / Model Scope	Main Contribution
[23]	2017	Image Classification (CNN)	Introduces translation-sensitivity maps and functions to visualize and quantify translation sensitivity in CNNs. Shows that they are not translation-invariant and demonstrates that data augmentation is the dominant factor to achieve it, outweighing architectural choices.
[31]	2023	Image Classification (CNN)	Proposes Alias-Free Convolutional Networks to achieve shift-equivariance and shift-invariance by eliminating aliasing. Tackles non-linear layers and downsampling as breakers of invariance and replaces them to achieve shift consistency.
[40]	2023	Image Classification (CNN)	Proposes the Scale and Rotation Equivariant Network, introducing a similarity convolution based on the Fourier-Argand representation that achieves general equivariance.
[42]	2011	Dataset Bias & Object Recognition	Provides a systematic analysis and comparison of dataset biases, identifying selection, capture, label and negative-set bias inherent to datasets, which makes them lack generalization for true visual understanding.
[17]	2021	Image Classification, Semantic Segmentation & Detection (CNN)	Demonstrates that global average pooling does not remove spatial information, showing absolute position can be encoded based on the ordering of channel dimensions and that CNNs rely on this mechanism.
[4]	2021	Image Classification (CNN)	Shows that most CNNs are not architecturally translation-invariant, but can learn it when pretraining on fully translated data. Demonstrates that pretraining data distribution is critical and fine-tuning can disrupt learned invariance.

Reference	Year	Task / Model Scope	Main Contribution
[9]	2023	Image Classification (ViT)	Analyzes why vision transformers lose shift-equivariance due to patch embedding, positional encoding, and subsampled attention. Proposes an adaptive polyphase algorithm to restore it and achieves improved results.
[30]	2020	Object Detection (CNN)	Introduces an evaluation metric to quantify shift equivariance in object detectors. Shows that modern detectors suffer from performance instability under shifts. Evaluates mitigation strategies and reports that all provide only partial improvements.

Chapter 3

Measuring, Analyzing, and Mitigating Translation Bias

This chapter encompasses the experimental work conducted within the scope of this dissertation. It details the experimental design, datasets, evaluation protocols, and methodological approaches followed to investigate positional and translation bias. A critical analysis of the obtained results is conducted, deriving empirical conclusions about the underlying mechanisms that cause and contribute to positional bias.

The experimental work is divided into two major components. Section 3.1 centers on the cork inspection dataset and explores its pair-defect setting, which allows for a practical study of detection consistency in industrial data. Sections 3.2 and 3.3 investigate the underlying mechanism behind the discovered asymmetries and examine mitigation strategies for decreasing said asymmetries.

3.1 Cork dataset and duplicate-defect setting

This section introduces a real-world industrial inspection scenario involving an analysis of images of cork material, in which defects must be correctly identified.

Within this context, a specific phenomenon is frequently observed: the same defect appearing twice in the same image. This scenario encompasses a practical task where translation bias can be rigorously investigated.

3.1.1 Dataset

The experiments conducted throughout this work incorporate a dataset of cork inspection images, in which each image corresponds to a single cork and may contain defects from 18 different classes, including "Open Insect Damage", "Crack", "Yellow Stain", and "Nail". As previously described, each image is composed of two regions: the left side shows the cork's cylindrical lateral surface unfolded onto a planar representation, while the right side presents the top and bottom views of the cork.

The dataset was created and collected for an industrial market task, under controlled conditions, resulting in a large and consistent set of images. The data is split into 10691 training images, 1336 for validation, and 1337 for testing.

This set of data is pertinent for this dissertation due to a recurring occurrence: the same defect may be seen twice in the same image at different vertical positions. This happens because during image acquisition, the cork rotates more than 360° , capturing about 20% of the lateral surface twice in the same image. If this section of the cork has any defects, duplicated visual instances of those defects appear, leading to cases like the one illustrated in Figure 3.1.

Effectively, this creates two virtually identical instances of the same defect and background per image, yielding a perfect case study for studying detection consistency under translation and analyzing how spatial shifts can lead to inconsistencies in detection, classification, and confidence.

In the test set, this duplication occurs in 433 instances, which are specifically used to evaluate the proposed consistency metrics, creating a rigorous and natural experimental setting that allows measuring:

- Intra-image consistency: the model should detect both instances as the same class and with similar confidence values
- Detection robustness: the model should detect both occurrences of the defect

The dataset provides a practical, controlled setting to study inconsistencies in detection performance under spatial translation. When evaluated with a standard YOLOv8 model trained on the dataset, many defect pairs, despite being visually identical, return differences in predicted classes, significant variations in confidence values, and, in some cases, complete detection failures, where one instance is detected and the other is missed.

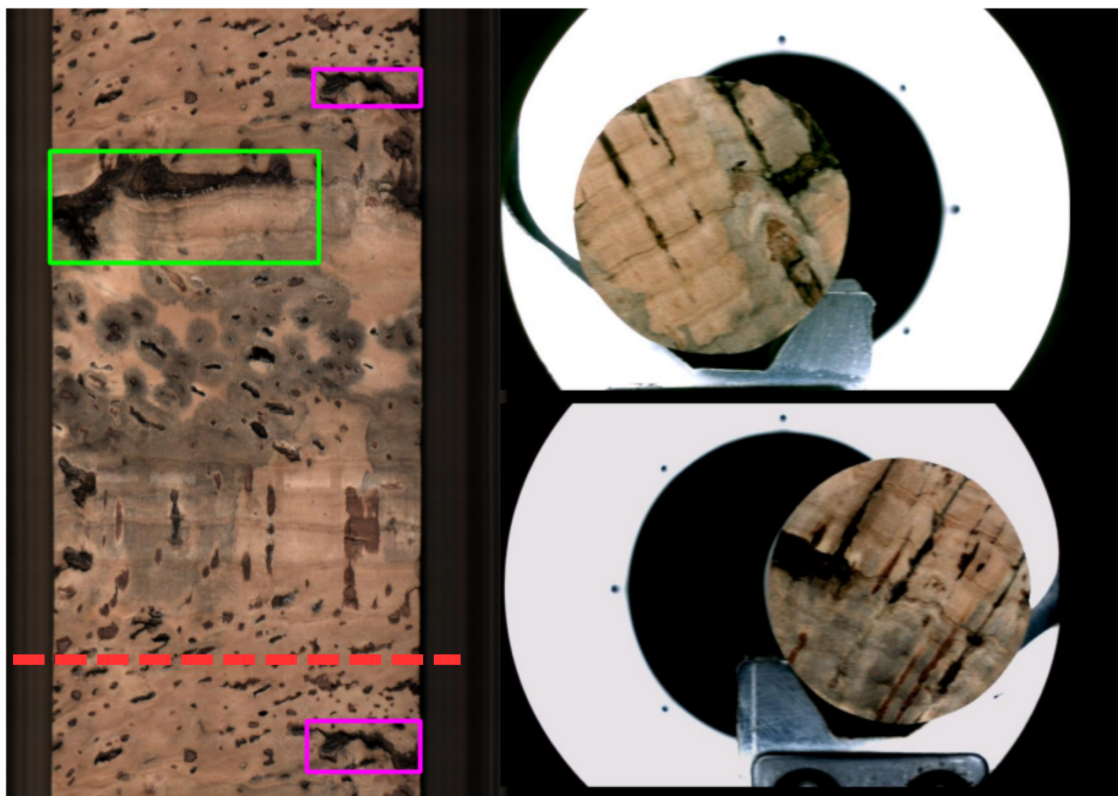


Figure 3.1: Left side of cork image shows a duplicated defect pair highlighted in pink. A singular (non-paired) defect is highlighted in green. The dotted red line shows when the cork rotates more than 360° and starts repeating visual content.

3.1.2 Evaluation Pipeline

To analyze the prediction consistency of the pair defects, an evaluation pipeline was developed. This pipeline was designed to quantify two main properties. The first is intra-image consistency, to measure if the model assigns the same class and similar confidence values to both instances of the pair defect. The second is detection robustness consistency, meaning whether the model is able to detect both instances, or if only one of them is detected.

The initial stage of the pipeline consists of identifying every instance of pair defects in the test set. Since the distance between duplicated defects is always the same (about 80% of the image's height when the 360° rotation starts repeating), only differing in their y-axis value, the duplicated defects can be accurately stored as pairs. This provides the target reference to be later used for comparing model predictions.

For a given trained detector, predictions are computed for the entirety of the test set. For every duplicated pair in an image, predictions are matched against the ground truth. This association is done based on the biggest **IoU** value found, using a fixed threshold of 0.50. If no valid prediction is found, the defect is considered undetected.

This initial matching is class-agnostic, being solely based on **IoU**, with the predicted class analysis being included later on. This allows quantifying localization and classification errors

separately.

Following this, information is extracted from each duplicated pair, such as:

- Is the first and/or the second instance of the defect detected?
- Do both predictions have the same class?
- What is the absolute difference between predicted confidence values?

Lastly, all duplicated pair metrics are aggregated to create global statistics that characterize how well the trained model deals with this phenomenon. These include:

- **Partial Miss Rate** ($\downarrow$): proportion of pairs in which exactly one of the elements is detected while the other is not. This is the principal metric because it works as a gating condition for the remaining consistency metrics.
- **Agreement Rate** ($\uparrow$): proportion of pairs (when both instances are detected) where the predicted class matches between instances and is correct. This metric also gates the Average Confidence Delta.
- **Average Confidence Delta** ($\downarrow$): average difference between the confidence values of the two correctly classified detections.

Additional statistics, such as **Total Miss Rate** ($\downarrow$) (out of all pair instances, how many have at least one defect of the pair missed), **Average Class Accuracy** ($\uparrow$), and **Average Confidence Mean** ($\uparrow$) (when both instances are detected), are computed to provide context and for general pair-only performance reporting.

This custom pipeline enables rigorous evaluation of a translation-sensitivity phenomenon that is not captured by traditional global metrics, focusing on model consistency and detection asymmetries.

3.1.3 Model Training and Mitigation Strategies

This section describes experimental work exploring different strategies to mitigate the phenomenon present in the corks dataset.

All runs are conducted with YOLOv8s, on Ultralytics 8.4.21, and for each model configuration, three different runs are executed to account for training randomness (with seeds 40, 41, and 42). YOLOv8 was selected as it is a widely adopted, well-documented modern detector, offering a representative architecture to study positional effects. The default YOLOv8 parameter configuration is utilized, and each model is trained for 30 epochs, with a patience of 10 (if for 10 epochs the validation metrics do not improve, the model stops training), achieving model convergence. The reported results, which are presented in Table 3.1, are the average across those three seeds.

The inference is done using a confidence threshold of 0.25 and a non-maximum suppression **IoU** of 0.45. All the models are evaluated using the same consistent conditions. To assess overall

performance, **mAP@50** for the left side of the image is reported. Training and inference are executed on a single Tesla V100-PCIE-16GB GPU.

3.1.3.1 Baseline

An initial default run was conducted in order to assess the baseline behavior of the model, achieving a **mAP@50** of 0.322. In this first run, most of the default train-time augmentations in the **YOLO** pipeline are disabled, besides the HSV color augmentations.

Regarding the pair-specific metrics, this model achieved an *Agreement Rate* of 0.843, a *Confidence Delta* of 0.099, and a *Partial Miss Rate* of 0.259. Overall, the baseline displayed a very high *Total Miss Rate* (0.583) and a low *Class Accuracy* (0.427). This initial configuration serves as a starting point, quantifying significant instability and asymmetry in detection between pair defects.

3.1.3.2 Default Runtime Augmentations

Enabling the default **YOLO** augmentations, specifically translation and scaling, led to a substantial improvement on both the model's general capabilities as well as the pair-specific metrics. This configuration yielded a **mAP@50** of 0.392, *Agreement Rate* rose to 0.871, *Confidence Delta* decreased to 0.086, and the *Partial Miss Rate* also improved to 0.229. Simultaneously, *Total Miss Rate* decreased from 0.583 (baseline) to 0.509, and *Class Accuracy* improved from 0.427 to 0.500.

As a whole, these results show that standard data augmentations can reduce translation bias by increasing generalization performance, though metrics like *Partial Miss Rate* still remain relatively high.

Every subsequent configuration will include these augmentations. Therefore, moving forward, these results will serve as the reference for all future comparisons.

3.1.3.3 Convolutional Layer Stride

Out of all the conducted architectural-only experiments, the strongest pair consistency result was obtained by changing the stride of the first convolutional layer from 2 to 1. These results, however, come at a higher computational cost: training time increases from about 1h25min to about 2h30min, and GFLOPS increase almost fourfold.

This model achieved an **mAP@50** of 0.396, only marginally higher than in the previous experiment. Nevertheless, it attained an *Agreement Rate* of 0.878, a *Confidence Delta* of 0.078, and a *Partial Miss Rate* of 0.199, totalling some of the best values among all approaches. *Total Miss Rate* also had a noticeable decrease to 0.466, and *Class Accuracy* increased to 0.534.

These results are particularly important because they show clear improvements on all three dimensions: the model had fewer cases of detecting one instance and missing the other, when both instances were detected, it attained a higher class *Agreement Rate*, and when the pair was correctly detected and classified, it showed a smaller confidence variation between the two instances. Remarkably, these improvements arise with only modest **mAP@50** gains, suggesting that global metrics may be misleading for a consistency problem like this one.

This conclusion suggests that pair defect consistency is strongly influenced by early spatial feature sampling. It demonstrates that reducing the initial downsampling, thereby preserving higher-resolution feature maps in the earliest stage of the network, greatly improves translation equivariance in detectors.

3.1.3.4 BlurPool layer

BlurPool, the custom layer presented in [44], was integrated into YOLO and tested as an alternative architectural strategy to improve shift-invariance using anti-aliased subsampling. However, the results it obtained were not competitive with those from other configurations.

The BlurPool model obtained a weak **mAP** of 0.349, which was lower than every other strategy (besides the first baseline), with an *Agreement Rate* of 0.873, *Confidence Delta* of 0.089, and a *Partial Miss Rate* of 0.238. Globally, the *Total Miss Rate* remained high (0.557) and *Class Accuracy* remained low.

Due to this module being originally developed for classification tasks, integrating it for object detection comes with hard trade-offs, such as less precise localization due to low-pass filtering.

These results show that, for this specific pair-consistency task, the trade-offs are not worth it and BlurPool is not a useful integration. They also highlight the need for developing detection-focused, invariant layers that can be effectively and broadly integrated into object detection architectures.

3.1.3.5 Large-Scale Data Augmentation

A large-scale data augmentation strategy was also evaluated, starting from the original $\approx 10k$ images and creating $\approx 30k$ more images.

The augmentation process generates 3 images for each original: one with a vertical flip, one with a horizontal flip, and one with both transformations. These operations are only applied to the left side of the image.

After this, a random circular translation on the y-axis is applied to the left side of each created image. The algorithm takes into account the location of the defects so that none are partially cropped. This step frequently destroys the pair-defect structure, in which defects are separated by about 80% of the image height, because each defect is near the top or bottom of the vertical strip. Hence, a circular translation usually shifts one of the defects to the opposite end of the vertical strip, placing it in a very close position to the other instance.

This strategy prioritizes image variability by deliberately ignoring the pair defect structure to understand whether more diverse data can improve overall model performance, and if so, if this improvement by itself leads to greater pair consistency.

This configuration produced the highest overall **mAP** so far at 0.400. It also achieved the highest *Agreement Rate* (0.908) and a competitive *Confidence Delta* (0.081). However, the *Partial Miss Rate* slightly worsened from the reference's 0.229 to 0.234, and the pair-only *Total Miss Rate* stayed high, at 0.514.

This experiment is particularly relevant because it demonstrates that better global detection does not directly imply improved translation robustness. Additionally, even though the model got better at pair consistency for class classification when both detections were present, it did not become more reliable at detecting both members of the pair as measured by *Partial Miss Rate*.

This suggests that increased generic data augmentations help the model learn what to predict. However, if the generated data does not contain the pair structure, when this structure appears in test images, the general gains may not translate into detecting both instances of the pair.

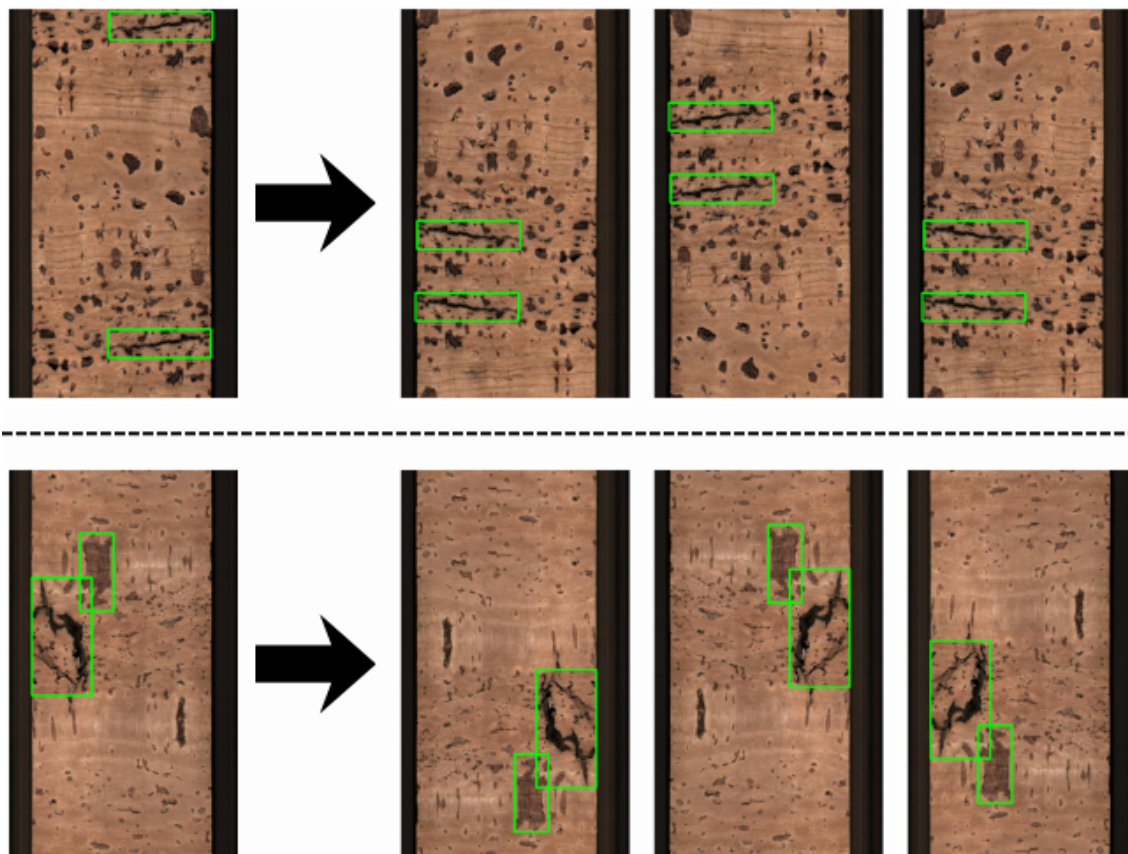


Figure 3.2: Two examples of the produced results after conducting the augmentation described in the Large-Scale Data Augmentation subsection. Each original image produces 3 augmentations. As exemplified in the top image, the pair defect structure is frequently destroyed. The right side of the cork stays untouched and is hidden in this Figure.

3.1.3.6 Oversampling Pair Defect Images

To address the limitation of the previous experiment, a more pair-defect-targeted strategy was tested. Out of the original $\approx 10k$ images, about 3k of them had at least one case of pair defects. Rather than simply using every image, augmentation focused solely on this subset, generating $\approx 9k$ additional examples.

Similar to the first step of the large-scale data augmentation, this process generates three additional images per original: one with a horizontal flip, one with a vertical flip, and one combining

both transformations, applied to the left portion of the image. This preserves the 80% vertical separation and keeps the pair defect structure intact, as illustrated in Figure 3.3.

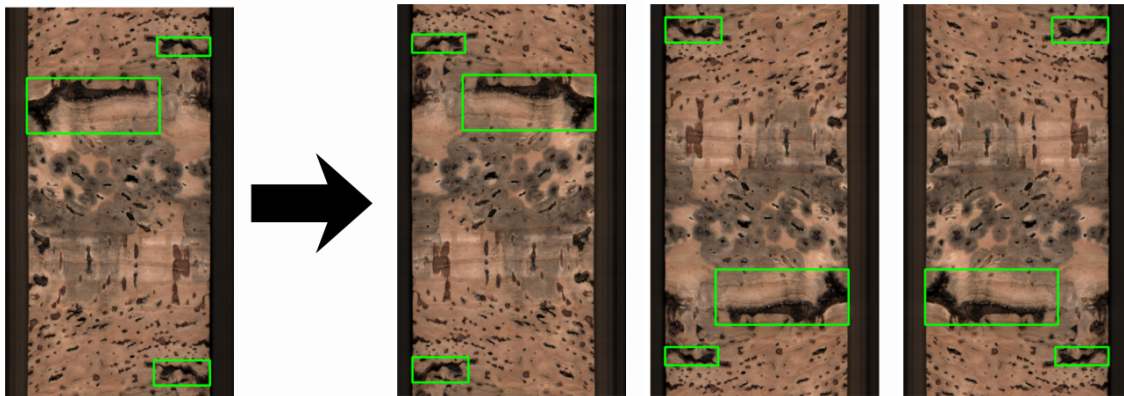


Figure 3.3: Two examples of the produced results after conducting the augmentation described in the Oversampling Pair Defect Images subsection. In contrast to the Large-Scale Data Augmentation pipeline, the pair defect structure is always preserved. The right side of the cork stays untouched and is hidden in this Figure.

The model achieved a $mAP@50$ of 0.389, an *Agreement Rate* of 0.886, a *Confidence Delta* of 0.082, and a *Partial Miss Rate* of 0.210. Additionally, even though the $mAP@50$ slightly decreased when compared to the reference or the large-scale augmentation configurations, metrics focused only on the pairs clearly improved: *Total Miss Rate* improved to 0.478 and so did *Class Accuracy* (0.533).

This contrast indicates that not all augmentation strategies are equally useful for mitigating the pair defect performance inconsistencies. It also demonstrates that structure-aware augmentation, which preserves the geometric relation of the pair defects, can be more effective than simply increasing data volume for metrics like *Partial Miss Rate*, even though metrics such as mAP are lower.

Despite not reaching the gains obtained using the stride-1 strategy, oversampling pair-defect images through a coherent and data-enriching process emerges as an attractive data-centric mitigation approach.

3.1.3.7 Conv0Stride1 + Oversampling Pair Defect Images

To assess whether mixing different strategies could yield even bigger benefits or if results would plateau, a strategy was evaluated that combines the two most effective mitigation strategies.

This configuration achieved the strongest $mAP@50$ of all experiments at 0.411, as well as the best *Partial Miss Rate* of the whole study at 0.168. Besides that, it obtained a very competitive *Confidence Delta* of 0.082 and *Total Miss Rate* of 0.428, significantly lower than any other configuration. This larger number of detections might have negatively influenced *Agreement Rate*, which slightly dropped to 0.867, an acceptable trade-off for a strong improvement of the *Partial Miss Rate*.

This experiment confirms that the combination of data-centric and architectural-focused mitigation strategies can be complementary, producing the best global score between consistency and general performance metrics.

Overall, this experiment displays that, in this work, the most effective pair mitigation strategy is one that simultaneously requires architectural intervention to reduce sensitivity to early downsampling, as well as a targeted, dataset-specific augmentation process that reinforces the pair structure.

3.1.3.8 Custom Loss Function

Lastly, two custom pair-consistency loss variants were developed. The first used a teacher-student dynamic in which the higher confidence anchor acted as a fixed teacher, with a Huber-penalized consistency term pushing the lower confidence pair toward it alongside another term which penalized class disagreement between the two. The second variation simplified this to a penalty on the absolute confidence gap for the ground-truth class, applied only when at least one anchor reached a confidence threshold. Both were added to the classification loss term with a gain factor.

A fundamental constraint of this approach was that geometric data augmentations could not be enabled, because it would break the matching algorithm between the fixed pre-annotated ground-truth pairs and the training predictions, which placed this strategy at a structural disadvantage relative to other configurations. Despite testing several hyperparameter combinations, the empirical results did not support its effectiveness.

These results suggest that forcing prediction consistency at the loss level may not be the best mitigation strategy for this particular scenario. It also emphasizes the increased difficulty and complexity of designing a custom loss function for a highly specific problem setting, which may ultimately return poor results.

3.1.4 Discussion

The experiments conducted on the cork dataset confirm that positional bias is not merely an abstract or controlled synthetic phenomenon, but also a practical issue with ramifications in real industrial data. Under the conditions of this dataset, where the exact same defect may appear twice in the same image, differences in classification, confidence, and detection outcomes can be interpreted as evidence of position-related inconsistencies.

The evaluation pipeline developed to quantify pair consistency showed that the baseline revealed substantial instability, revealing information that could not be assessed using only global measures such as **mAP**. Across all tested strategies, the results confirm that clear improvements in pair consistency can be achieved, and that some mitigation approaches outshine others. In Figure 3.4, four examples are given using two of the best mitigation strategies that show unequivocal cases of improved pair consistency in predictions.

Standard runtime augmentations produced a visible improvement over the baseline, suggesting data diversity can help reduce positional bias. However, some of the strongest reductions in

asymmetric results, leading to smaller *Partial Miss Rates*, were obtained by changing the stride of the first convolutional layer from 2 to 1, reinforcing the idea that reducing early downsampling may be a critical factor for mitigation.

The data-centric experiments clarify the relationship between generalization and positional robustness. Large-scale augmentation that often destroyed the pair structure increased overall **mAP@50** and *Agreement Rate* but failed to improve robustness in detecting both instances of a pair. By contrast, the structure-aware oversampling, which preserved pair geometry, reduced the *Partial Miss Rate* even though it reports a weaker **mAP@50**.

These results support the idea that augmentation should align with the structure of the bias being addressed, and that when the problem is rooted in a specific spatial bias, structure-focused augmentation can lead to better performance than generic data enrichment.

The mixed strategy combining Conv0Stride1 with structure-preserving oversampling achieved the overall best results, with the most significant reduction in *Partial Miss Rate* and the highest **mAP@50**. The combination of the two strategies leads to complementary benefits, demonstrating that both architectural and data-related factors can be employed together for effective mitigation.

The BlurPool experiment also provides relevant conclusions. In a defect detection scenario, the smaller sensitivity to precise feature details that this layer incorporates by design led to performance deterioration. This result reinforces the need for more research in detection-specific architectural solutions rather than adaptation of classification-oriented modules.

The custom-loss experiment highlights the practical difficulty of enforcing consistency through loss optimization, showing that a theoretically attractive strategy can still fail.

As a whole, these experiments show that positional bias is a concrete source of inconsistencies in practical settings. Among the tested approaches, reducing early downsampling and structure-preserving data augmentations yielded the best outcomes. These conclusions strengthen the overall argument of this dissertation that translation robustness must be studied with task-specific metrics and addressed through complementary mitigation strategies compatible with the nature of the problem.

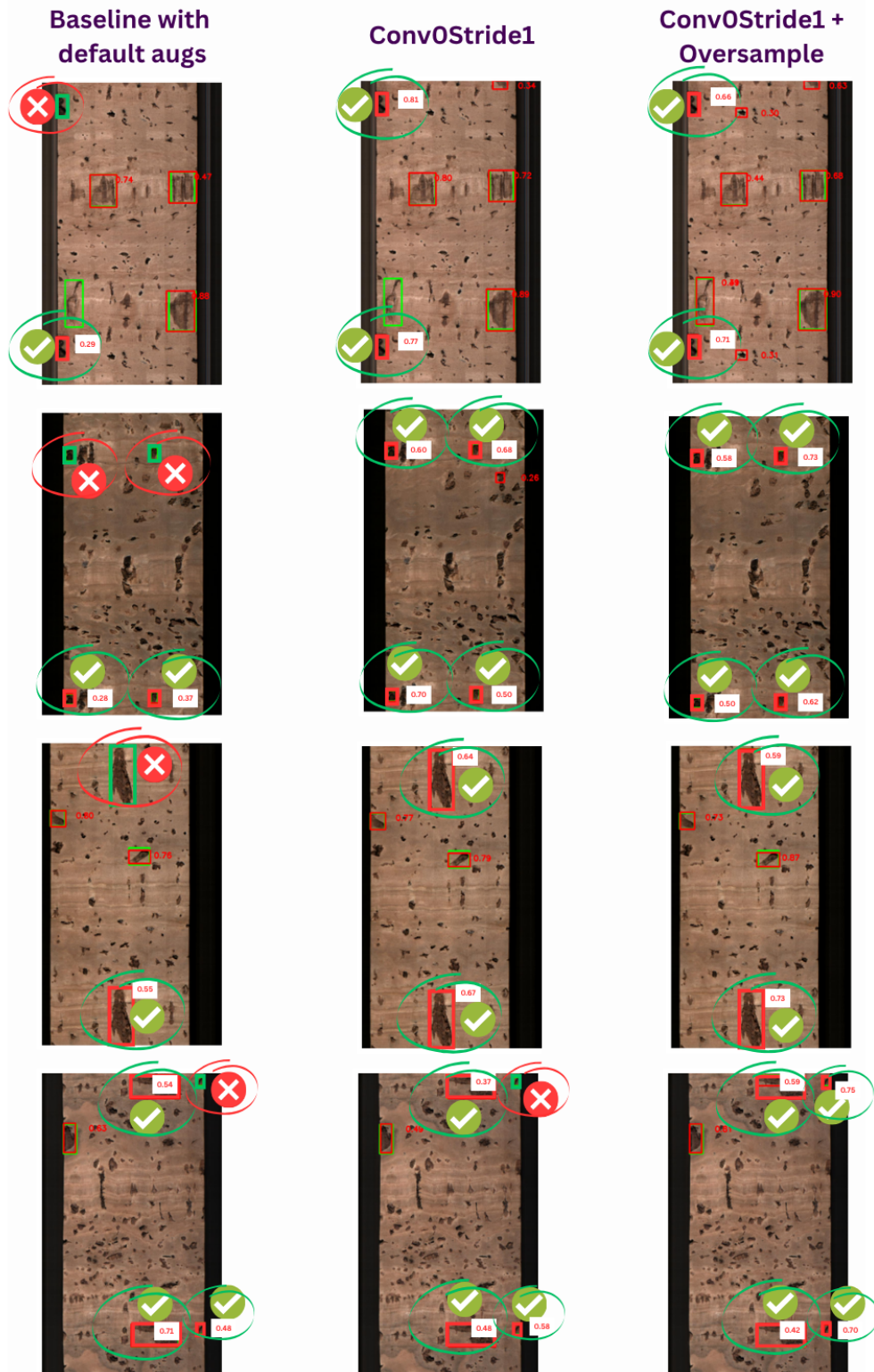


Figure 3.4: Four images being detected using different training strategies. The green bounding boxes show the ground-truth annotations, while the red displays the predicted bounding box and confidence of said prediction. The right part of the image is hidden for easier visualization. The first model, the baseline with runtime augmentations analyzed in 3.1.3.2, always misses one of the instances of a given pair defect in the example images, while the Conv0Stride1 and Conv0Stride1 + Oversample models (presented in 3.1.3.3 and 3.1.3.7) have more consistent pair detections. The cross and correct mark icons display if a pair defect was correctly detected or not. Some images contain defects/predictions of non-pair-defects.

Table 3.1: Pair Consistency Metrics Across Different Strategies

Configuration	Partial Miss ↓	Agreement ↑	Conf. Δ ↓	mAP@50 ↑	Conf. Mean ↑	Total Miss ↓	Class Acc. ↑
Baseline	0.259	0.843	0.099	0.322	0.610	0.583	0.427
Runtime Augs (RT)	0.229	0.871	0.086	0.392	0.633	0.509	0.500
Conv0Stride1	0.199	0.878	0.078	0.396	0.646	0.466	0.534
BlurPool	0.238	0.873	0.089	0.349	0.611	0.557	0.449
Large-Scale Augs	0.234	0.908	0.081	0.400	0.598	0.514	0.519
Pair Oversampling	0.210	0.886	0.082	0.389	0.624	0.478	0.533
Mixed (Oversample + Conv0Stride1)	0.168	0.867	0.082	0.411	0.630	0.428	0.558

3.1.5 Influence of defect size on pair consistency

As supplementary analysis to the previously presented mitigation experiments, an additional study was conducted to evaluate how prediction consistency for pair defects varied with defect scale.

This experiment divides all pair defects into quartiles, and utilizes the inferior quartile Q1 (smallest defects) and the superior quartile Q4 (biggest defects), as seen in Figures 3.5 and 3.6, to verify how heterogeneous detection is across scales and to analyze how different strategies can reduce or amplify said variations.

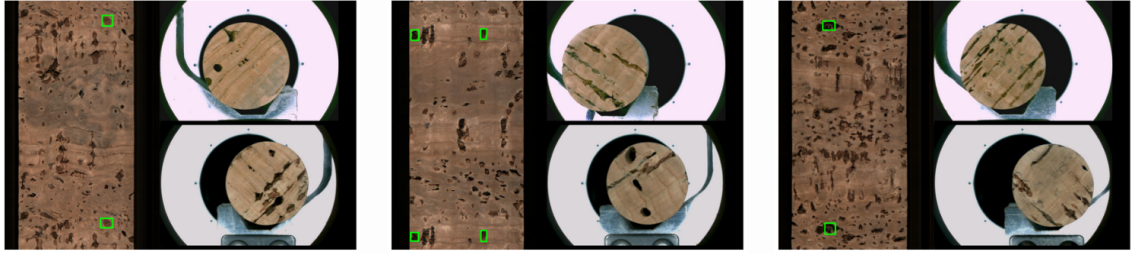


Figure 3.5: Three examples of pair defects, with their bounding boxes highlighted in green, which belong to Q1

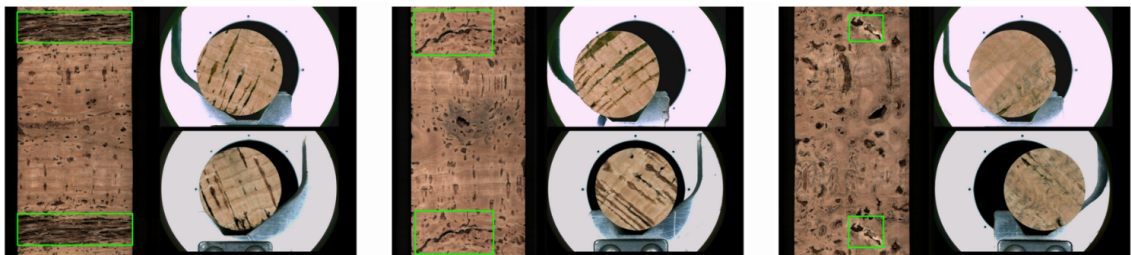


Figure 3.6: Three examples of pair defects, with their bounding boxes highlighted in green, which belong to Q4

The principal focus of this analysis was again *Partial Miss Rate*, since this metric captures the major issue this phenomenon causes: situations in which one defect is detected but its pair is not. Metrics such as *Total Miss Rate*, *Confidence Delta*, and *Confidence Mean* are also observed for supporting further conclusions.

Table 3.2: Duplicate defect consistency by defect scale.

Configuration	Group	Partial Miss↓	Agreement↑	Conf. Δ↓	Conf. Mean↑	Total Miss↓	Class Acc.↑
RT	Q1 (small)	0.387	0.954	0.099	0.531	0.742	0.315
Pair Oversampling	Q1 (small)	0.333	0.911	0.087	0.530	0.698	0.340
Conv0Stride1	Q1 (small)	0.284	0.890	0.075	0.543	0.664	0.358
Mixed	Q1 (small)	0.256	0.914	0.091	0.545	0.604	0.429
RT	Q4 (large)	0.166	0.864	0.077	0.698	0.333	0.640
Pair Oversampling	Q4 (large)	0.153	0.863	0.073	0.690	0.316	0.659
Conv0Stride1	Q4 (large)	0.165	0.857	0.077	0.681	0.347	0.608
Mixed	Q4 (large)	0.119	0.826	0.078	0.665	0.286	0.657

This section analyzes four representative configurations for the previous section: the base reference with runtime augmentations, pair-structured oversampling, stride-1 on the first convolutional layer, and the mixed strategy that combines the previous two. Table 3.2 reports the obtained results.

The first noteworthy, although expected, result is that, looking at the reference RT configuration, smaller defects show substantially worse overall and inconsistency values when compared to big defects. *Partial Miss Rate* is 0.387 for Q1, but only 0.166 in Q4. This same pattern continues in *Total Miss Rate*, which reaches 0.742 for smaller defects while staying at 0.333 for larger ones. Besides this, *Confidence Mean* also displays significantly lower values for small defects (0.531 for Q1 and 0.698 for Q4). Together, these results clearly demonstrate that the consistency issue in pair defects is significantly stronger at smaller scales, where accurate and stable detection relies more heavily on fine-grained spatial information.

An examination of the oversampling strategy reveals an improvement in both groups: for the smaller defects, the *Partial Miss Rate* improves from 0.387 to 0.333, and in the big ones, from 0.166 to 0.153. Additionally, *Total Miss Rate* also shows gains in both cases. Looking at these results, the relative differences in performance between small and big defects remain similar, as both groups exhibit comparable proportional reductions (14% for small defects and 8% for large defects). This suggests that the oversampling strategy helps the model in absolute terms, but does not significantly decrease the vulnerabilities seen in small defects.

In contrast, the Conv0Stride1 configuration returns an entirely different behavior. The gains obtained from this strategy come virtually all from the smaller defects: *Partial Miss Rate* decreases from 0.387 to 0.284, a relative reduction of almost 27%, while for the big defects it remains practically unchanged (0.166 to 0.165). The same pattern follows for the *Total Miss Rate*, which improves from 0.742 to 0.664 in Q1, but even slightly increases in Q4. This asymmetry is particularly informative because it suggests that the change in stride of the initial layer disproportionately impacts smaller defects, where the loss of spatial details is more critical. This finding supports the interpretation proposed in the previous subsection: a significant portion of the inconsistencies in pair defects arises from spatial downsampling, which leads to the loss of critical spatial detail, especially affecting smaller defects, which show greater sensitivity to such information loss.

Lastly, the mixed strategy produces improved results for both groups. In small defects, it

reaches a *Partial Miss Rate* of 0.256, and for big defects, a reduction to 0.119, alongside a *Total Miss Rate* decrease to 0.604 in Q1 and 0.286 in Q4.

This analysis demonstrates that consistency for pair defects strongly depends on defect scale. As a whole, smaller defects exhibit significantly higher *Partial Miss Rate*, *Total Miss Rate*, along with lower confidence mean and class accuracy, exhibiting less consistent detection. This result supports the hypothesis that positional variability becomes more problematic when the instance is small and therefore contains less spatial detail, making it more susceptible to information loss during downsampling.

The comparison between strategies reinforces how delaying early downsampling mainly improves smaller defects, while oversampling pair instances improves performance more uniformly. The mixed strategy reaffirms that combining improved fine-grained spatial detail and exposition to relevant examples is the most effective mitigation for both big and small defects.

3.2 Periodic Confidence Modulation

The consistency issue observed in pair defects is indicative of a broader limitation present in CNN-based object detection models: imperfect translation equivariance. Therefore, to better understand the impact of positional shifts on detection stability, a controlled experiment was designed to isolate the effect of vertical translations on model predictions.

3.2.1 Experimental Protocol

To systematically evaluate the impact of object position on detector behavior, particularly regarding the confidence assigned to predictions, a vertical treadmill sweep experiment was conducted using YOLO models trained on the full cork dataset.

For each selected image, the left region (corresponding to the planar unwrapped lateral surface of the cork) was vertically translated by increments of 1 pixel across all valid positions, excluding those where the target object would be partially cropped. The selected images had only a single defect in the left region for easier matching and controlled variability. This process generates a sequence of images in which the defect remains visually identical, as illustrated in Figure 3.7, differing only in its vertical position. This procedure was repeated across multiple images to ensure robustness of the observations.

This experimental setup is grounded in prior work on translation invariance, which shows that subsampling operations (e.g. strided convolutions) can break invariance. As a result, this may lead to significantly different outputs for shifted versions of the same input, except when translations are aligned with the network’s sampling stride.

For each translated sample, inference is performed, and the prediction corresponding to the target object is selected as the highest-IoU correctly classified bounding box produced, subject to a minimum IoU threshold of 0.25 with respect to the ground-truth.

The focus of this analysis is not overall detection performance, but rather consistency under translation. Therefore, only defects that are detected across the entire set of translated images

are considered. This ensures a recall of 100% (i.e., the target is detected at all shifted positions), isolates confidence variation from detection failures, and enables a controlled investigation of potential detection patterns.

After obtaining the resulting sequence of confidence scores for all vertically translated samples, periodic behavior is quantified using z-normalized sliding-window autocorrelation over the reported confidence values across integer lags. Since each step corresponds to a 1-pixel shift, the lag values directly map to spatial displacements, allowing precise recognition of periodic patterns.

Images are warped to square dimensions to control for aspect-ratio effects on inference scaling.

Lastly, an architectural ablation study was conducted to determine the origin of any periodic structures found, particularly focused on the role of downsampling operations.

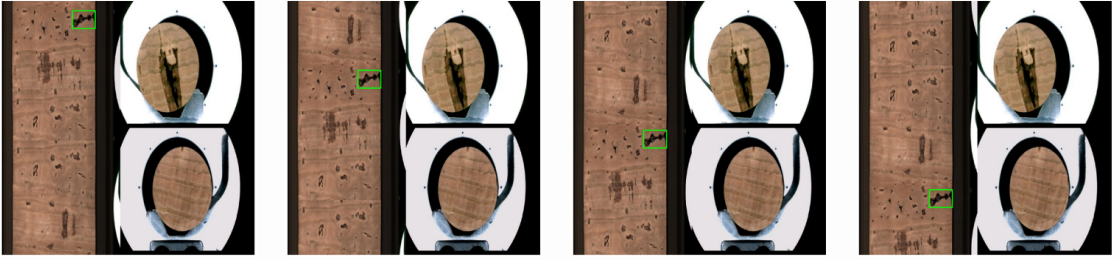


Figure 3.7: Examples of 4 translations from the entire treadmill sweep. The left region of the image, including the defect (green box), is translated pixel by pixel for all valid positions.

3.2.2 Autocorrelation Analysis

To determine whether the confidence signal exhibits a periodic structure as the object is translated vertically, the confidence values produced by predictions are analyzed as a one-dimensional signal indexed by the target’s pixel displacement.

Let c_t denote a given confidence value assigned to the target object located at vertical translation step t , where consecutive values correspond to 1-pixel shifts. Given the fact that each translation step is one pixel, a lag of value L is numerically equivalent to a vertical displacement of L pixels.

A windowed self-similarity measure is used to identify repeating patterns over a segment. Given a confidence sequence $c = \{c_1, c_2, \dots, c_N\}$, overlapping windows of a fixed length W are extracted:

$$w_i = [c_i, c_{i+1}, \dots, c_{i+W-1}], \quad i = 1, \dots, N - W + 1 \quad (3.1)$$

where the index i indexes the position within the confidence sequence and N is the length of the confidence sequence.

In this work, a window length of $W = 120$ is used for all experiments, providing a sufficiently large local context while preserving sensitivity to spatial patterns.

Each window is then z-normalized to remove possible variations in mean confidence, ensuring similarity reflects the pattern shape itself rather than absolute values:

$$\tilde{w}_i = \frac{w_i - \mu_i}{\sigma_i} \quad (3.2)$$

where μ_i and σ_i are the mean and standard deviation of window w_i

For a given lag L , the similarity between two windows separated by L positions (i.e., L pixels) is computed as the mean elementwise product between their normalized versions:

$$r_i(L) = \frac{1}{W} \sum_{j=1}^W \tilde{w}_{i,j} \tilde{w}_{i+L,j} \quad (3.3)$$

Because each window is z-normalized, this quantity is equivalent to the correlation between two local confidence patterns. The final autocorrelation score for a given lag L is obtained by averaging this value across every valid window pair:

$$R(L) = \frac{1}{M_L} \sum_{i=1}^{M_L} r_i(L) \quad (3.4)$$

where M_L is the number of valid window pairs for that lag.

The analysis is repeated for every valid integer lag (at least one window pair exists), with respect to the confidence signal length.

Following this pipeline, results can be interpreted directly: a strong peak at lag 32, for instance, indicates that the confidence signal tends to repeat itself every 32 pixels of vertical translation.

This measuring procedure provides a direct way to identify and quantify periodic detection patterns in confidence variation.

3.2.3 Existence of Structured Periodicity

Experiments were conducted using YOLOv8s with default inference settings in Ultralytics 8.4.21, which sets the inference image size to 640. To evaluate the effect of spatial scaling, the same images were warped to create five squared resolutions: 320×320 , 640×640 , 960×960 , 1200×1200 , and 1280×1280 . For each resolution, a full vertical treadmill sweep dataset was generated following the procedure explained in Subsection 3.2.1.

Each dataset was created by first warping the original images to the target resolution and then applying all valid 1-pixel vertical shifts, excluding positions that would crop the target object plus a little margin. As a result, for an image of size $N \times N$, the sweep covers all valid vertical positions within that resolution.

Across all datasets, the confidence signal showed a clear, stable periodic structure. This indicates that the variation in confidence does not change randomly with object position, but instead follows a consistent and repeatable spatial pattern.

Most importantly, the discovered periodic patterns in YOLOv8s varied systematically with the input resolution. As illustrated in Figure 3.8 and summarized in Table 3.3, the same image resized

to different sizes shows distinct periodic signals. For the test resolutions, the following empirical relationship was observed:

$$\text{Pattern} = 32 \cdot \left(\frac{D}{640} \right) \quad (3.5)$$

where D is the dataset image size used to generate the 1-pixel shift sweep.

Table 3.3: Dataset image sizes and reported periodic pattern

Image Size	Periodic Pattern
320x320	16
640x640	32
960x960	48
1200x1200	60
1280x1280	64

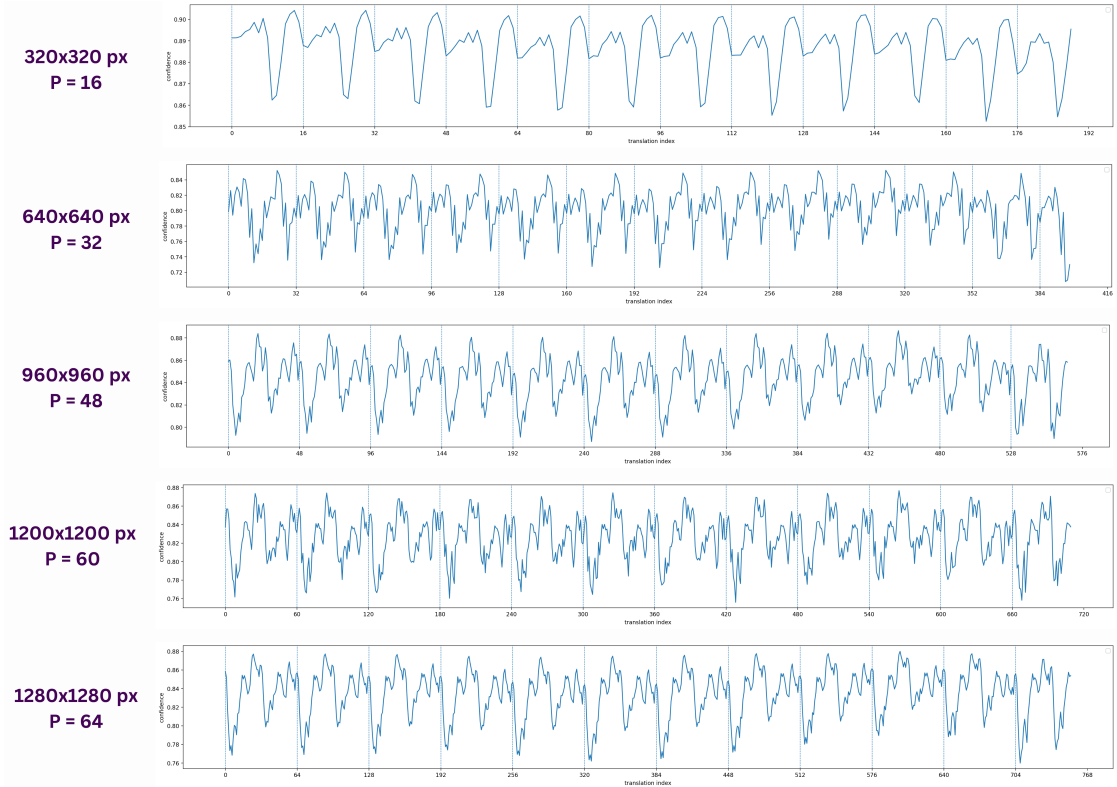


Figure 3.8: Confidence sequence for the same image at five resolutions, with default inference $imgsz = 640$. The dominant period scales linearly with the image size.

This relationship strongly suggests that the observed periodicity is driven by the detector’s spatial stride since the factor of 32 matches the maximum effective stride of YOLO’s backbone.

This means that the structured period observed is primarily governed by the architecture’s down-sampling geometry rather than other possible image-specific factors.

When the defect is correctly detected for all vertical positions, pattern correlation consistently obtained very high values, frequently over 0.95 similarity, across all image sizes, providing evidence of strong and stable periodicity.

Confidence Variation and Mean Coupling

To further illustrate the observed behavior, Figure 3.9 presents the confidence trajectory for a representative dataset. The confidence fluctuates across positions, but after completing the full period, it returns to its exact confidence value at pixel 32 as it was at pixel 0 (0.946). This precise absolute value recurrence occasionally occurs, particularly among high-confidence datasets, and provides direct evidence of a strict periodic detection. However, this is not the case for every dataset.

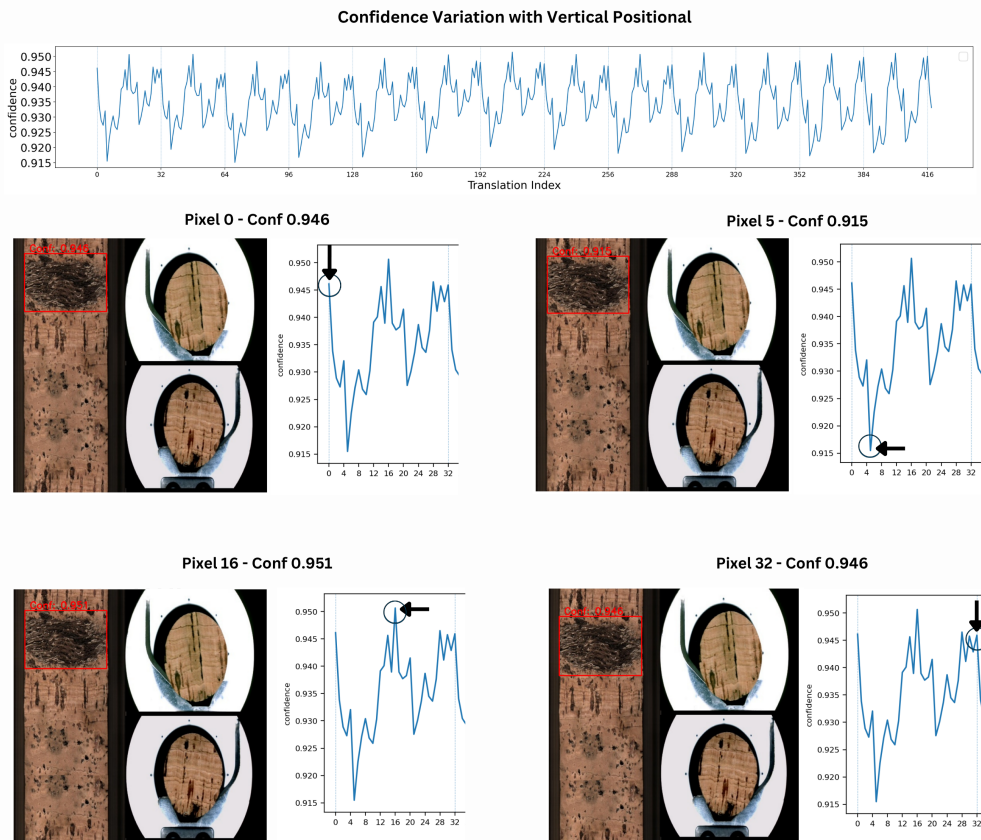


Figure 3.9: Example of confidence variation and corresponding pixel shifts for a high-confidence detection. Notably, the confidence at pixel 32 returns to the exact value (0.946) observed at pixel 0.

While the periodic structure itself always remains stable, the absolute confidence that follows that pattern may vary within a single dataset. Figure 3.10 illustrates this behavior. As can be seen, the 32-pixel periodicity is consistent throughout the entire sweep in both cases, yet the

absolute confidence values float upwards significantly. This suggests that the context in which an object exists and its absolute location in the image can also influence its prediction confidence. Nonetheless, this behavior does not interact destructively with the periodic component.

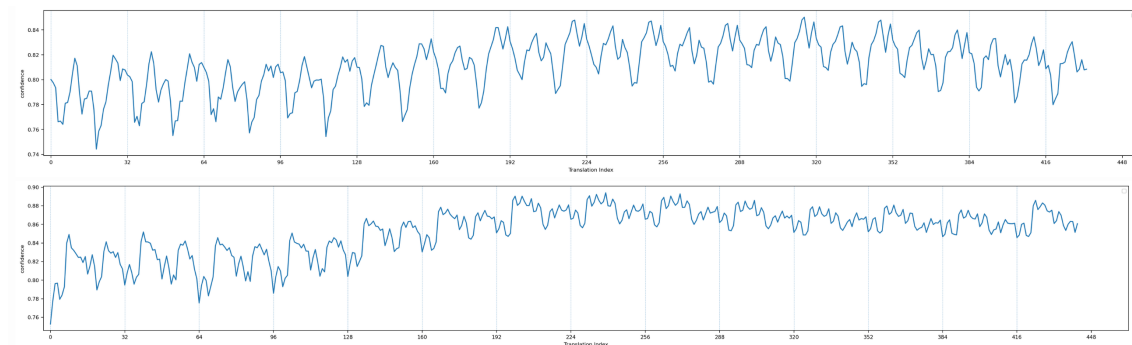


Figure 3.10: Confidence trajectories for two datasets displaying large absolute confidence variations alongside the 32-pixel periodic structure. In both cases, the stride-induced periodicity is maintained.

Another relevant relationship was found among the 640×640 datasets with perfect recall. Mean confidence is strongly and negatively correlated with confidence standard deviation (Pearson $r = -0.73$, Spearman $\rho = -0.80$), as illustrated in Figure 3.11, meaning datasets with lower mean confidence have bigger confidence oscillations, and vice-versa. It is important to note that this coupling has practical consequences for detection reliability. Since low mean confidence datasets are naturally closer to the minimum confidence threshold, their larger confidence oscillation amplitude means that their periodic patterns are more likely to push individual translations below the detection threshold. In other words, the same periodic structure that generally causes minor confidence fluctuations in high-confidence datasets can more frequently lead to recall failures in low-confidence ones.

This relationship is heteroscedastic because the variance in confidence fluctuation is substantially higher at low mean confidence, and decreases sharply at approximately mean confidence of 0.7. A possible explanation for this relationship is that lower-confidence detections correspond to weaker feature activations that are more sensitive to spatial shifts, which may lead to larger fluctuations in confidence values. In contrast, high-confidence detections arise from stronger feature responses that remain more stable under positional changes. Notably, after z-normalization, the strength of the 32-pixel periodicity remains largely unchanged across confidence levels, indicating that periodicity is a property of the architecture and completely distinct from the absolute confidence level.

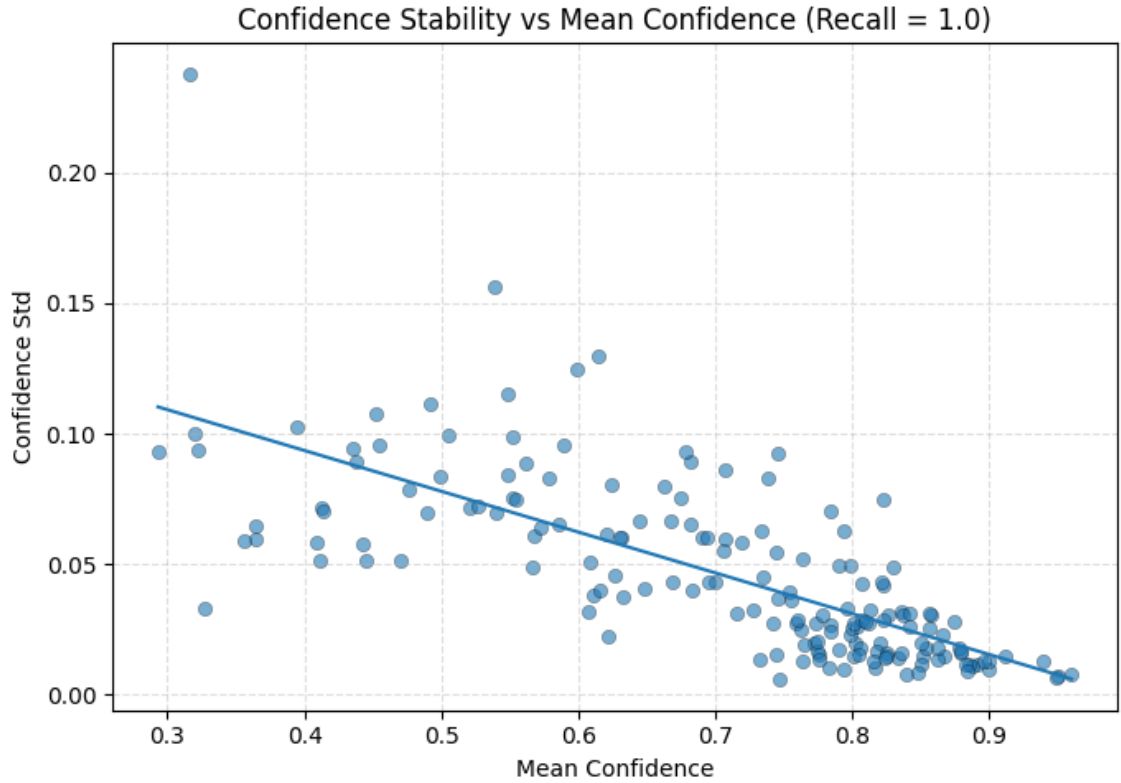


Figure 3.11: Mean confidence versus confidence standard deviation across 171 perfect-recall 640×640 treadmill datasets. The negative trend (Pearson $r = -0.73$, Spearman $\rho = -0.80$) indicates that lower-confidence detections show greater variability.

Generality across model size, version and family

For the 640×640 datasets, the reported dominant period was 32 pixels. To verify this behavior was not specific to YOLOv8s, additional tests were conducted using this dataset across model scales and architectural variants. Different YOLOv8 model sizes (nano and medium) were evaluated, and the same 32-pixel period emerged, showing the effect does not depend on model size.

Subsequently, the same test was expanded to newer installments of the YOLO architecture. YOLOv11 [20], which remains architecturally very similar to YOLOv8, displayed the same pattern. YOLOv26 [21], which supports an NMS-Free, end-to-end inference pipeline, also showed the same 32-pixel period, indicating that Non-Maximum Suppression is not responsible for the observed structure.

Lastly, RT-DETR [45] was also tested using the same datasets. Although this detector integrates a transformer detector, it still relies on multi-scale feature maps produced by a CNN backbone with the same spatial resolutions as YOLO. Again, the same resulting periodic behavior dominated, suggesting the effect is not tied to a specific detection head mechanism, but rather to the underlying multi-scale spatial hierarchy.

3.2.4 Scale Covariance and Periodic Families

Further experiments showed that the inference size with which YOLOv8 performs predictions is a major factor determining the period observed in the confidence signal. When the same vertically shifted datasets are evaluated at different inference resolutions `imgsz`, the resulting pattern length changes systematically.

Across all evaluated cork-image treadmill datasets, for a dataset generated from images of size $D \times D$, the expected base period is empirically given by:

$$P_0 = 32 \cdot \left(\frac{D}{\text{imgsz}} \right) \quad (3.6)$$

where D is the dataset image size used to generate the 1-pixel shift sweep, and `imgsz` is the inference resolution used during prediction.

This defines the base predicted period P_0 . In this context, scale covariance means that the characteristic confidence pattern that emerges can be mapped proportionately to the variation in the scale of the input image and the inference resolution. When the inference resolution changes, the periodic pattern also rescales.

The behavior was evaluated across treadmill datasets generated from robustly detected cork defects, and is present on all tested dataset sizes (320x320, 640x640, 960x960, and 1280x1280) for a variety of `imgsz`.

Period Family and Harmonic Dominance

As shown in 3.4, in the majority of experiments P_0 matched the dominant observed period found; however, this isn't always the case. Instead, the dominant observed period can sometimes fall under a harmonic of P_0 . Additionally, subharmonics of P_0 can occasionally show high periodic results and even exceed it in individual images, but in the executed experiments, none were ever consistently stronger than P_0 across images. To represent this, the observed periodic structure can be represented as a family of admissible periods derived from P_0 :

$$F(P_0) \in \{kP_0 \mid k \in K\}, \quad K \subseteq \left\{ \frac{1}{2}, 1, 2, 3, 4, \dots \right\} \quad (3.7)$$

In these experiments, the dominant observed period was very frequently P_0 itself, but in some cases it corresponded to a larger harmonic.

To avoid ambiguity, P_{dom} is used to denote the dominant observed period, which is the smallest lag that consistently shows the stronger peaks across images for a given input size and inference resolution.

For certain inference resolutions, particularly under strong rescaling, the reported average confidence decreased, sometimes even leading to missed detections. Since the goal is to quantify spatial periodicity, we restrict the analysis to datasets where the target was detected at all translation shifts (recall = 1.0).

As shown in Table 3.4, P_0 is the dominant peak in the majority of combinations. The dominant periods typically yielded pattern similarity values above 0.95, with some exceptions falling slightly

lower, confirming the strength and stability of the periodic structure across inference resolutions. One of the three datasets used for testing, generated from a smaller, more challenging defect, sporadically exhibited lower similarity values (between the 0.70 - 0.90 range), particularly in the 320x320 datasets. This accounts for the increased inter-dataset variability observed in those cases.

Even though subharmonics of P_0 occasionally showed high similarity values, particularly $P_0/2$, they were generally weaker and did not appear consistently across images. No consistent relationship was found between the properties of those defects.

Notably, when the $\frac{D}{\text{imgsz}}$ ratio is simple (e.g., $\times 1$, $\times 2$, $\div 2$) the P_{dom} always equaled P_0 , which also frequently occurs when $\text{imgsz} < D$. In contrast, certain resize ratios (particularly when $\text{imgsz} > D$) sometimes show harmonic-dominated behavior, where P_0 is noticeably lower than at its harmonics. This effect happened in the following three examples:

- 640 dataset at $\text{imgsz} = 1024$: $P_0 = 20$, yet, as can be visually perceived in Figure 3.12, the spectrum is dominated by $4P_0 = 80$, while the periodicity at 20 and 40 exists but is noticeably weaker.
- 960 dataset at $\text{imgsz} = 1024$: $P_0 = 30$, yet, as can be visually perceived in Figure 3.13, the spectrum is dominated by $8P_0 = 240$, while the periodicity at 30 exists but is noticeably weaker.
- 960 dataset at $\text{imgsz} = 1280$: $P_0 = 24$, yet, as can be visually perceived in Figure 3.14, the spectrum is dominated by $2P_0 = 48$, while the periodicity at 24 exists but is noticeably weaker.

In these three example configurations harmonics of P_0 dominated over P_0 itself, though P_0 still remained detectable with medium-to-high correlation values. This harmonic dominance was never observed (in the test cases) when image size and inference size were the same. For this reason, a hypothesis is that this behavior reflects interactions between non-integer resize ratios introducing interpolation artifacts. Critically, the dominant periods all remained multiples of P_0 , indicating that the stride lattice itself seems to be preserved.

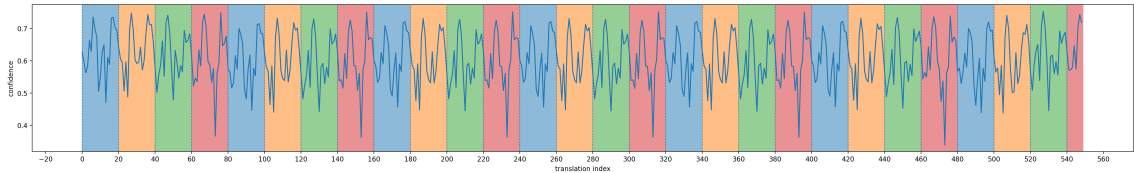


Figure 3.12: Confidence variation for an image in 640x640 resolution size inferred at $\text{imgsz}=1024$. Colors are used to improve the visualization of long patterns.

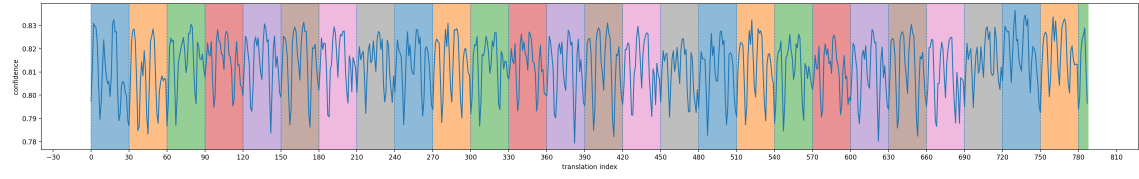


Figure 3.13: Confidence variation for image in 960x960 resolution size inferred at `imgsz=1024`. Colors used for improving visualization for long patterns.

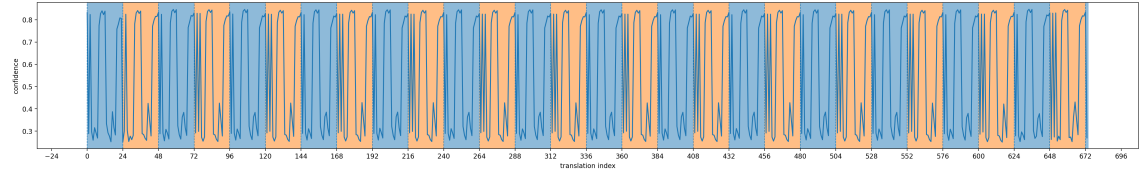


Figure 3.14: Confidence variation for image in 960x960 resolution size inferred at `imgsz=1280`. Colors used for improving visualization for long patterns.

Table 3.4: Predicted Base Period P_0 from Equation 3.6 versus the dominant observed period P_{dom}

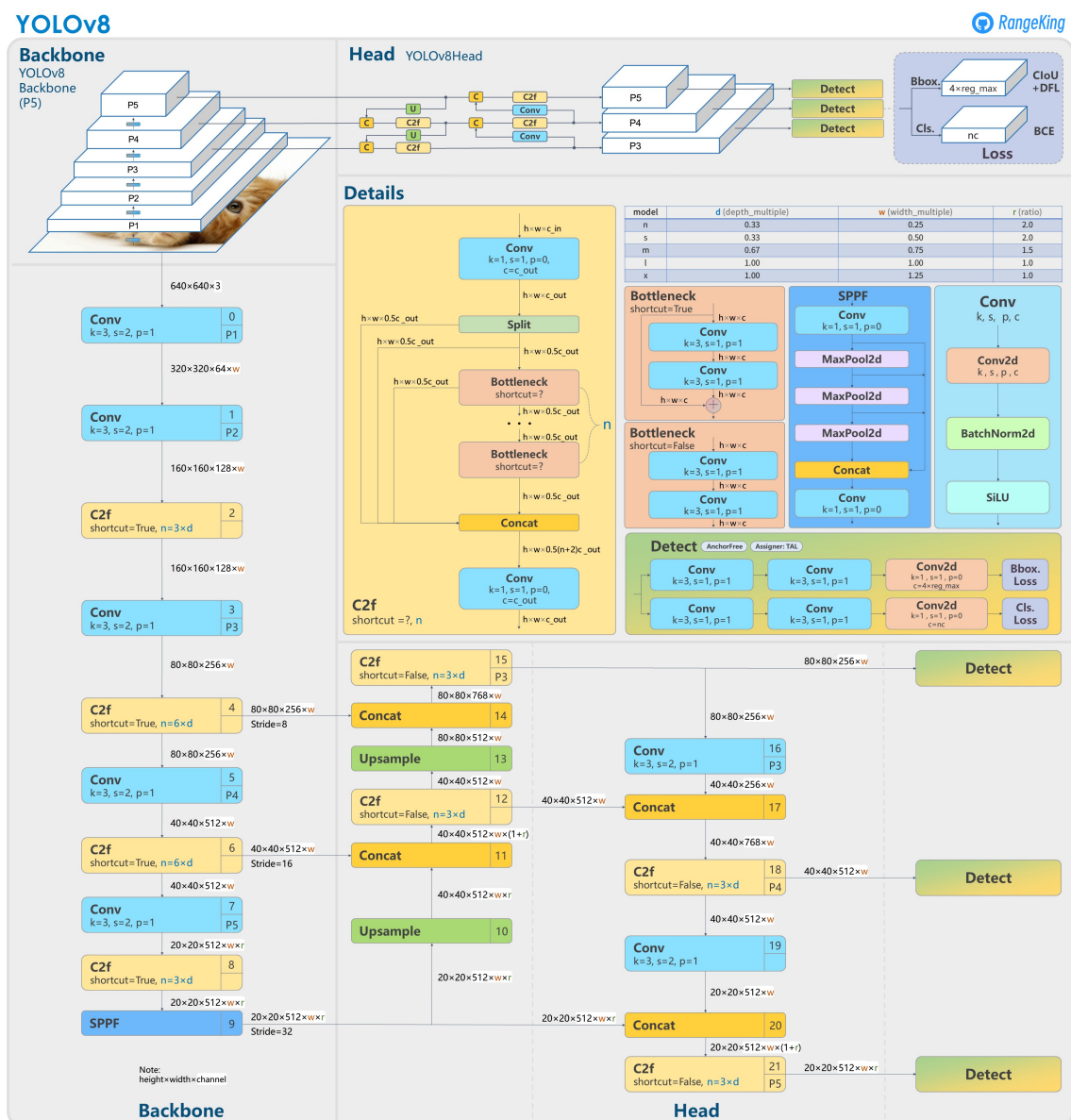
ImgSize	InferenceSize	P_0 / P_{dom}	Avg / Median Similarity for Actual
320x320	256x256	40 / 40	$0.82 \pm 0.18 / 0.90$
320x320	320x320	32 / 32	$0.91 \pm 0.10 / 0.96$
320x320	512x512	20 / 40	$0.82 \pm 0.13 / 0.88$
320x320	640x640	16 / 16	$0.95 \pm 0.06 / 0.98$
640x640	256x256	80 / 80	$0.90 \pm 0.11 / 0.95$
640x640	320x320	64 / 64	$0.94 \pm 0.07 / 0.97$
640x640	512x512	40 / 40	$0.93 \pm 0.05 / 0.95$
640x640	640x640	32 / 32	$0.97 \pm 0.03 / 0.98$
640x640	1024x1024	20 / 80	$0.93 \pm 0.07 / 0.96$
640x640	1280x1280	16 / 16	$1.00 \pm 0.00 / 1.00$
960x960	256x256	120 / 120	$0.90 \pm 0.11 / 0.95$
960x960	320x320	96 / 96	$0.95 \pm 0.06 / 0.98$
960x960	480x480	64 / 64	$0.94 \pm 0.05 / 0.97$
960x960	640x640	48 / 48	$0.98 \pm 0.01 / 0.99$
960x960	960x960	32 / 32	$0.99 \pm 0.00 / 0.99$
960x960	1024x1024	30 / 240	$0.94 \pm 0.02 / 0.93$
960x960	1280x1280	24 / 48	$0.99 \pm 0.01 / 0.99$

ImgSize	InferenceSize	P_0 / P_{dom}	Avg / Median Similarity for Actual
1280x1280	256x256	160 / 160	$0.93 \pm 0.08 / 0.97$
1280x1280	320x320	128 / 128	$0.96 \pm 0.04 / 0.98$
1280x1280	512x512	80 / 80	$0.97 \pm 0.05 / 0.99$
1280x1280	640x640	64 / 64	$0.98 \pm 0.02 / 0.99$
1280x1280	1024x1024	40 / 80	$0.95 \pm 0.04 / 0.96$
1280x1280	1280x1280	32 / 32	$0.97 \pm 0.03 / 0.98$

3.2.5 Architectural Origin: Dependence on Maximum Effective Stride

Like many **CNN**-based object detectors, YOLO-based detectors follow a three-stage architecture composed of a backbone, a neck, and a detection head, shown in Figure 3.15 (where the neck and detection head are grouped under a "head" block for simplicity).

YOLO's backbone uses successive strided convolutions, reducing spatial resolution and producing feature maps at strides of 8, 16, and 32 in relation to the input image's dimensions. Therefore, the maximum effective stride is defined as the coarsest feature-map resolution present in the backbone, which is a grid that is 32 times coarser than the original image. The neck fuses the three aforementioned multi-scale representations using a feature pyramid structure, enabling detection across object scales. Despite multi-scale fusion, the largest feature map (P5, stride 32) remains embedded in the hierarchy, making it an inherent component of the feature hierarchy.

Figure 3.15: YOLOv8 architecture diagram provided by <https://github.com/RangeKing>

After establishing that the default YOLOv8 model exhibits a stride-32 periodicity in confidence predictions, a set of architectural experiments was conducted to analyze the significance of the coarsest available stride in YOLOv8 for determining the emerging periodic pattern.

(A1) Remove the P5 detection head and keep the backbone.

The first architectural ablation was to remove the P5 (stride 32) branch from the detection head completely, while keeping the backbone unchanged and retraining the model. Importantly, the model still computes the backbone stride-32 stage, which propagates through the neck via the feature pyramid network, where it is upsampled and fused into the P4 (stride 16) and subsequently the P3 (stride 8) pathways.

Under this modification, the results exhibited the same 32-pixel pattern as previously seen.

The persistence of the same pattern led to the conclusion that the P5 head is not its main driver, indicating that the continued presence of stride-32 features within the backbone is a key factor.

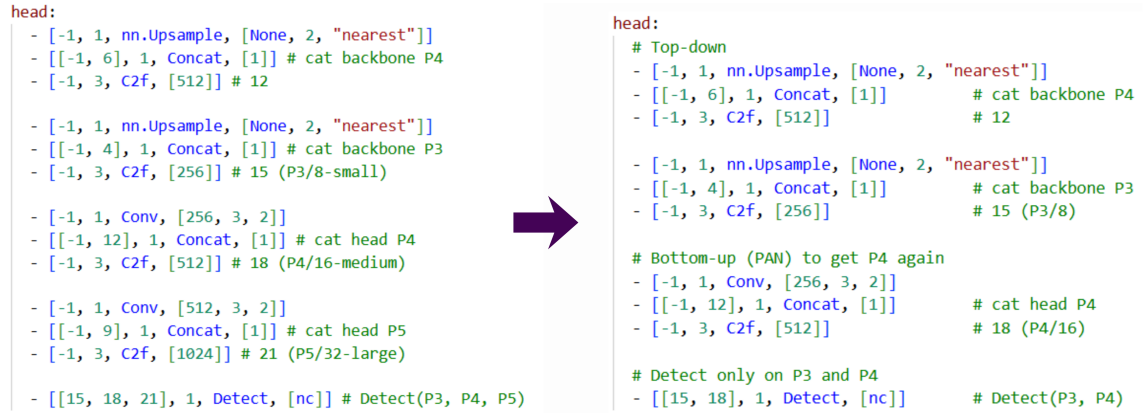


Figure 3.16: YAML configuration of the detection head for the default architecture (left) and the A1 ablation (right). The P5 detection branch and its connection to the Detect layer are removed, while the backbone remains unchanged.

(A2) Remove the stride-32 stage entirely.

A second architectural ablation completely removed the stride-32 stage from the backbone and all associated pathways. This modification effectively changed the maximum effective stride of the network from 32 to 16, meaning that no stride-32 feature map is ever produced or fused into the model.

This configuration shifted the periodic pattern from 32 pixels to 16 pixels. Such behavior is exactly what is expected if the periodicity is driven by the coarsest sampling stride present in the feature hierarchy.

This outcome provides strong evidence that the 32-pixel periodicity found in the previous experiments originates from the stride-32 sampling lattice present in the backbone-neck pipeline. It also demonstrates that when the stride-32 features are completely removed, and the coarsest feature map halves to stride-16, the periodicity correspondingly halves to 16 pixels.

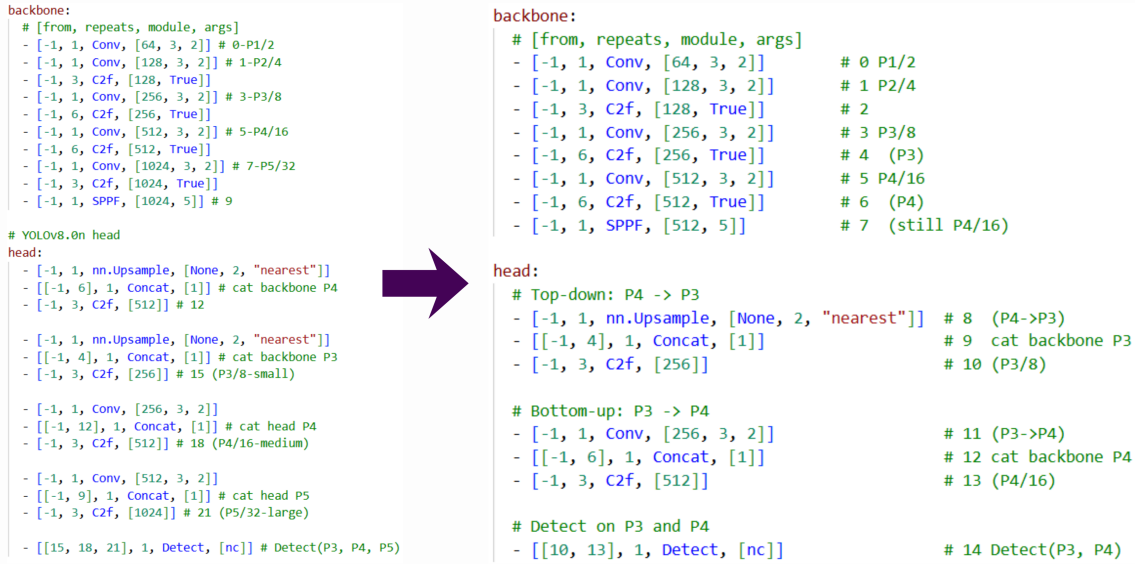


Figure 3.17: YAML configuration of the default architecture (left) and the A2 ablation (right). The entire stride-32 backbone stage is removed together with the P5 detection branch, reducing the maximum effective stride from 32 to 16.

(A3) Conv0Stride1.

In the final experiment, the default YOLOv8 architecture was retained, but the initial convolutional layer was modified to use a stride of 1 instead of 2. This change removes the first factor-of-2 spatial downsampling from the network, functionally shifting all feature maps by one level. As a result, the Detect head operates on feature maps with strides [4,8,16] instead of the original [8,16,32].

Using this design, the periodicity shifted from 32 pixels to a clean dominant 16-pixel pattern again. Such behavior is consistent with the hypothesis that the periodic pattern is governed by the largest effective sampling stride present in the model.

Together with the previous ablation (complete removal of stride-32 components), this experiment further supports that the observed periodicity, regardless of detection head, is governed by the backbone’s downsampling structure and its propagation through the feature pyramid.

```

backbone:
# [from, repeats, module, args]
- [-1, 1, Conv, [64, 3, 2]] # 0-P1/2
- [-1, 1, Conv, [128, 3, 2]] # 1-P2/4
- [-1, 3, C2f, [128, True]]
- [-1, 1, Conv, [256, 3, 2]] # 3-P3/8
- [-1, 6, C2f, [256, True]]
- [-1, 1, Conv, [512, 3, 2]] # 5-P4/16
- [-1, 6, C2f, [512, True]]
- [-1, 1, Conv, [1024, 3, 2]] # 7-P5/32
- [-1, 3, C2f, [1024, True]]
- [-1, 1, SPPF, [1024, 5]] # 9

backbone:
# [from, repeats, module, args]
- [-1, 1, Conv, [64, 3, 1]] # 0-P1/1
- [-1, 1, Conv, [128, 3, 2]] # 1-P2/4
- [-1, 3, C2f, [128, True]]
- [-1, 1, Conv, [256, 3, 2]] # 3-P3/4
- [-1, 6, C2f, [256, True]]
- [-1, 1, Conv, [512, 3, 2]] # 5-P4/8
- [-1, 6, C2f, [512, True]]
- [-1, 1, Conv, [1024, 3, 2]] # 7-P5/16
- [-1, 3, C2f, [1024, True]]
- [-1, 1, SPPF, [1024, 5]] # 9

```

Figure 3.18: YAML configuration of the default YOLO backbone (left) and the A3 ablation (right). The only modification is the stride of the first convolutional layer, which is reduced from 2 to 1. This changes the maximum stride from 32 to 16.

Overall, these experiments lead to the conclusion that the confidence with which an object is detected is strongly influenced by its phase relative to the model’s internal sampling grid, as illustrated in Figure 3.19. These phases repeat periodically and originate from the sampling lattice induced by strided downsampling and its propagation through the multi-scale feature hierarchy.

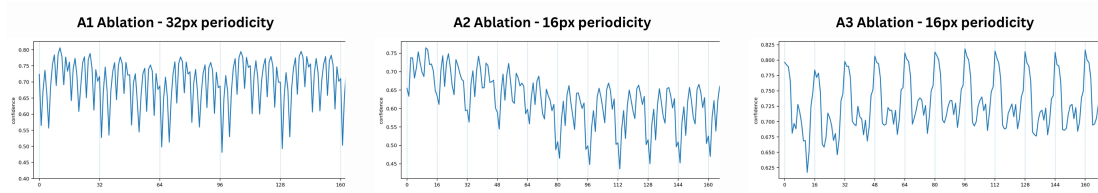


Figure 3.19: Lag correlation plots for the same treadmill dataset across the three ablations. In A1, the 32 pixel periodicity remains intact. In A2 and A3, the dominant period shifts to 16 pixels.

3.2.6 Cross-Dataset Validation and Periodic Confidence Structure

Representative experiments were conducted on a one-class subset of a second dataset [7] with substantially different characteristics in order to assess cross-dataset generality regarding the observed periodic behavior.

This second dataset differed from the first in that it contained smaller defects and was, in general, easier to detect, with the model achieving near-perfect detection performance.

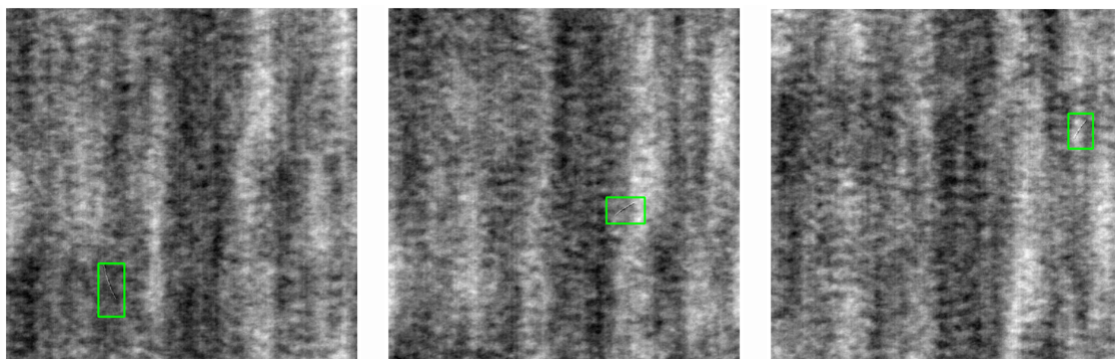


Figure 3.20: Example images of the cross-dataset used for validation, defect boxes highlighted in green

After applying the same experimental pipeline used for the original dataset, periodic patterns were consistently observed across the vertical transformations. In the majority of cases, the expected dominant periodicity patterns appeared, indicating that the previously found confidence structure remained intact. Notably, the dominant observed period P_{dom} corresponded to $P_0/4$, showing extremely high correlation values. This confirms that the phenomenon is not tied to a specific dataset. However, it raises the question of why the emerging periodicity consistently manifests as a sub-harmonic of the periodicity values found in the first dataset under identical size configurations.

3.2.6.1 Defect Size Does Not Determine the Dominant Sub-Harmonic

A natural first hypothesis for this variation is that smaller defects would be handled by the high-resolution P3 detection head (stride 8), medium defects by P4 (stride 16), and larger defects by P5 (stride 32), with the corresponding stride determining the observed periodicity. In the cork dataset, defects span a wide range of sizes but exhibited a constant periodicity at lag 32 (using 640×640 images and inference size of 640). Regardless of size, defects did not switch their dominant lag to 16 or 8, confirming that defect size in itself is not the main driver of periodicity values. The same conclusions followed after analyzing the second dataset, where constant periodicity values emerged in spite of defect size, though not at lag 32 but at lag 8 instead. Defects from the second dataset (with a bounding box dimension close to 1% of the image’s area) exhibited strong stride-8 periodicity, while similarly sized defects in the cork dataset retained the stride-32 periodicity. These results indicate that defect size is not the source of the observed variation.

3.2.6.2 Training Distribution Influences Periodic Behavior

Subsequent experiments led to a deeper understanding of the detected behavior. The explanation lies in how the training distribution interacts with the model’s stride hierarchy. Defect size and appearance determine which stride levels meaningfully contribute to predictions, effectively selecting the dominant period. Coarser strides that receive fewer positive signals for a given distribution progressively contribute less to the final detection.

Further evidence was provided causally through a controlled training experiment using the second dataset. To demonstrate this, a pipeline was developed that uniformly zoomed by a factor of 2x every image in the training and validation sets, preserving the defects' relative location but increasing their area fourfold (Figure 3.21). This experiment used images with 640×640 dimensions and an inference size of 640. The test set remained the same.

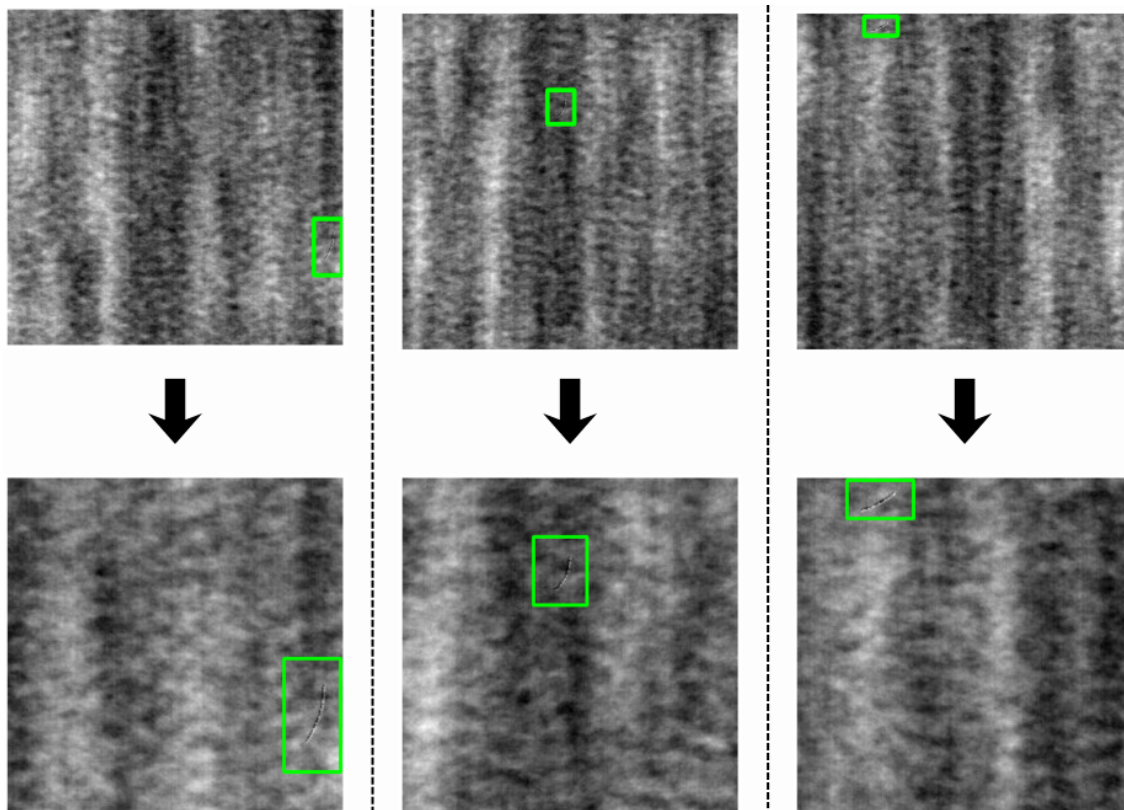


Figure 3.21: Examples of the conducted zoom-in transformation. The top row includes images from the original dataset, and the bottom row displays their zoomed-in versions.

A model retrained on this zoomed dataset, when tested on the original untouched test set, showed a substantial reduction in the previously seen stride-8 periodicity strength.

Using the retrained model, lag-8 correlation dropped from approximately 0.99 (results from the model trained on the initial dataset) to values between 0.66 and 0.84 across the same test images, while lag-16 and lag-32 remained at near-perfect correlation. Since the only variable across the two experiments was the training and validation data, this demonstrates that the dominant periodicity is a property of the trained model. This indicates that the characteristics of the training data influence which stride pathways the model relies on most, leading to the emergence of the corresponding periodicity.

To illustrate some effects caused by this mechanism and better visualize it, this study analyzed the classification branch of each detection head by applying a sigmoid activation to the logits, yielding per-cell confidence maps that reveal which heads actively contribute to detection. As shown in Figures 3.22 and 3.23, the model trained on the original dataset relies predominantly

on the P3 head, while P4 and P5 exhibit negligible activation. When trained on zoomed-in data, the P4 head became substantially more active, while the activation of P3 decreased. Notably, in Figure 3.23 P3 still maintains a stronger peak activation than P4. Nevertheless, in both cases, the periodicity shifts consistently: the lag-8 decreases substantially (from 0.99 to 0.66 and 0.77, for Figure 3.22 and Figure 3.23, respectively) while lag-16 becomes the dominant period.

Taken together, these results indicate that the dominant harmonic period emerges from the stride pathways the model has learned to rely on. The corresponding head activations serve as a partial observable indicator of that reliance: retraining on zoomed-in data shifts both the periodicity and the head logits together, though head activation alone does not fully determine the period.

This suggests that the composition of the training data directly influences how the model distributes detection and spatial responsibilities across its stride hierarchy which in turn determines the observed dominant period under translation.

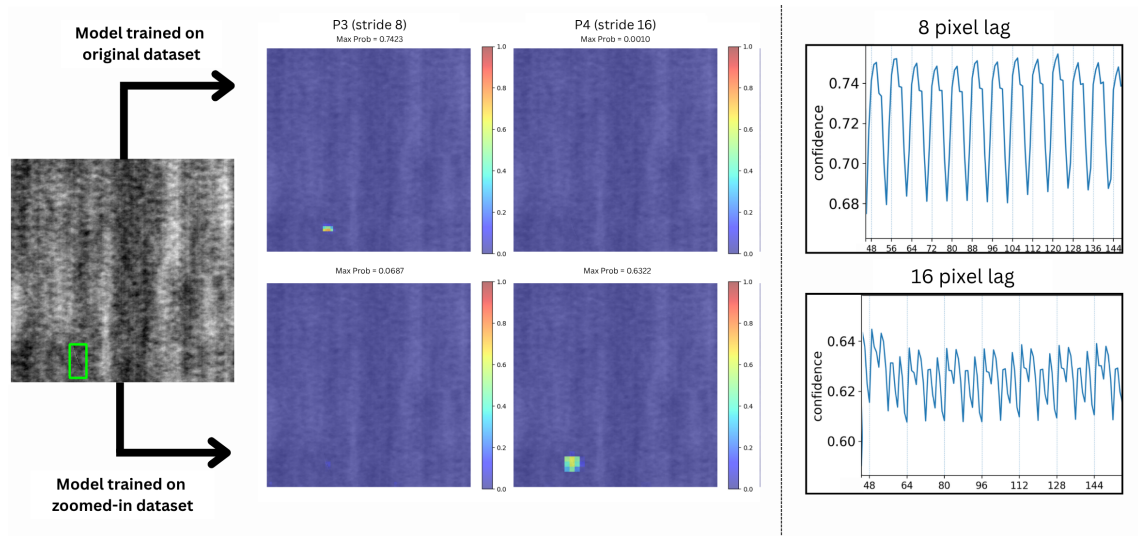


Figure 3.22: Left: Per-head classification activation for example image: original model (top) vs zoom-in model (bottom). Detection shifts from P3 (stride 8) to P4 (stride 16). P5 is silent in both and is omitted. Right: corresponding autocorrelation curves shifting from lag 8 to lag 16 after retraining.

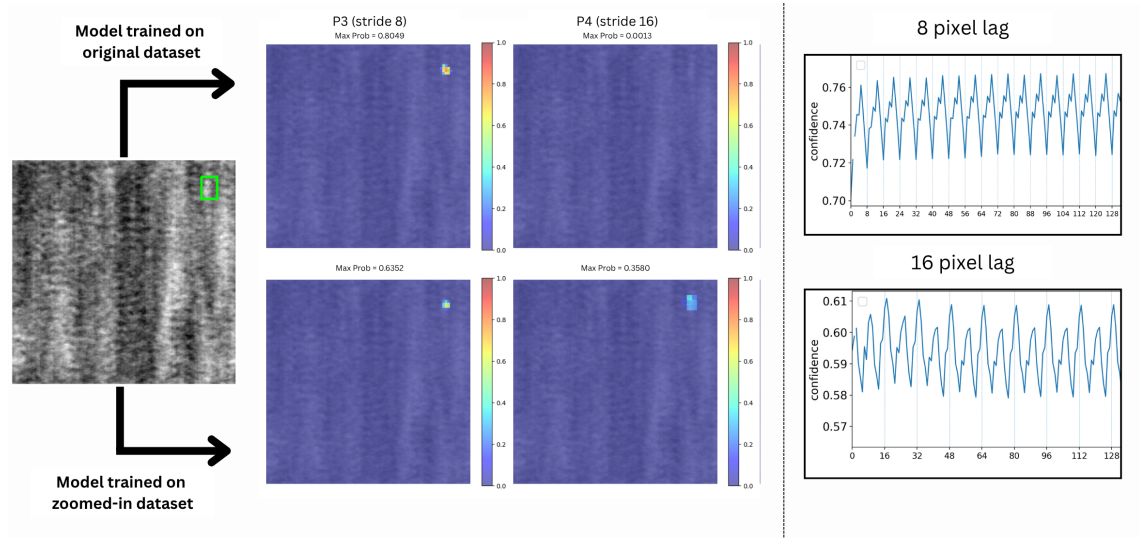


Figure 3.23: Left: Per-head classification activation for example image: original model (top) vs zoom-in model (bottom). P4 emerges from silence while P3 remains dominant, P5 is silent in both and is omitted. Right: corresponding autocorrelation curves shifting from lag 8 to lag 16 after retraining.

3.2.6.3 Generalization Across Datasets

The findings regarding the periodic variation of confidence in predictions under translation generalize across datasets, despite substantial differences in detection setting, defect appearance, and training data composition. Both datasets exhibit this variation, and both have a dominant period that follows the formula defined in Equation 3.6, though the base stride factor is not necessarily 32 as originally observed, but rather corresponds to the stride of the training-dominant pathway.

Based on the gathered results, we can combine the resolution scaling with the training distribution findings to generalize the formula in Equation 3.6 to

$$P_0 = s_{\text{train}} \cdot \frac{D}{\text{imgsz}} \quad (3.8)$$

where s_{train} is the stride of the dominant pathway that emerges depending on the distribution of the training data. For the cork inspection dataset, which has many defects with varying sizes per image, $s_{\text{train}} = 32$. For the second dataset, which possesses a single small defect per image, $s_{\text{train}} = 8$. We can empirically conclude that YOLOv8 CNN architecture defines the stride lattice and the used resolutions, together with training data, determine where the dominant period falls within that lattice.

3.2.7 Interpretation and Implications for Positional Bias

The experiments conducted throughout this section demonstrate the existence of position-dependent architectural bias in object detectors, expressed through resolution-dependent periodic patterns that structurally influence detection confidence under target translations. The collected evidence

strongly supports that this phenomenon is driven by the network’s internal sampling lattice, which emerges from strided downsampling and propagates through the backbone-neck hierarchy. Additionally, the dominant harmonic of this confidence oscillation is directly related to the stride pathways that the trained model relies on, which are strongly dependent on the training data’s composition.

Although these findings indicate that detection performance depends on an object’s placement relative to the model’s internal grid, the precise mechanism that sometimes makes absolute confidence higher or lower between phases is still unresolved in this work.

Another empirical observation is that confidence variation for all the vertical positions in a given treadmill sweep dataset is strongly related to the average prediction confidence for that dataset. Mean confidence is negatively correlated with confidence standard deviation, meaning lower-confidence datasets exhibit larger absolute confidence fluctuations across translations. Nevertheless, after z-normalization, the periodic structure remains preserved.

These findings contextualize and help explain the inconsistency problem found in the pair defect setting studied in this dissertation. If detector confidence depends on spatial position, then two visually identical defects appearing at different image positions may predictably lead to different outcomes. The conducted experiments show how an architectural source of positional bias can translate into unreliable detection for practical scenarios.

Overall, these results confirm that positional bias in modern detectors is not merely a data artifact but a structural component embedded in the feature extraction and model training process.

3.3 Mitigating Confidence Variability

The previous section established that vertically translating the same target object leads to systematic variations in detector confidence. The objective of the following experiment is not to remove the underlying structure embedded in YOLO’s architecture that causes the periodic fluctuations, but to evaluate how different training or architectural strategies can reduce the magnitude of the confidence variation seen across shifted positions.

3.3.1 Experimental Protocol

The same treadmill sweep protocol previously described was used. Using a test set of 273 images, each was warped to 640×640 , and the left unwrapped cork region was circularly translated by applying 1-pixel vertical shifts across all valid positions. For each shifted sample, inference was done at a 0.15 confidence threshold. After inference, predictions were matched with the ground-truth target using an IoU threshold of 0.25. Since the objective of this study is to measure confidence variation rather than achieve only final high-confidence predictions, this relatively low threshold was used to avoid discarding too many uncertain predictions.

Four configurations developed and presented in Section 3.1 were evaluated, as well as an additional combination:

- Baseline, corresponding to the default YOLOv8 configuration with runtime augmentations. (Subsection 3.1.3.2).
- BlurPool, which utilizes the blurpool layer for anti-aliasing before downsampling operations. (Subsection 3.1.3.4).
- Conv0Stride1, which reduces the stride of the first convolutional layer from 2 to 1. (Subsection 3.1.3.3).
- Large-Scale Data Augmentations, corresponding to the mass data augmentation strategy (Subsection 3.1.3.5).
- Conv0Stride1 + Large-Scale Data Augmentations, which combines these two previously mentioned approaches.

The Large-Scale Data Augmentations strategy is used instead of the oversampling strategy from Section 3.1, as the latter specifically targets the pair-defect structure, which is not relevant for this translation setting, while the first one focuses on enriching data diversity more generally.

Three different seeds (40, 41 and 42) were used to train each configuration, ensuring fairness and reliable results. The analysis was divided into two parts. Firstly, detection robustness was measured for all 273 treadmill datasets and compared between strategies.

Subsequently, confidence stability was measured only for datasets in which perfect recall was achieved, i.e., those in which the target object was successfully detected at every shifted position. For direct comparisons between strategies, common perfect-recall subsets are computed. These subsets restrict the analysis to samples where all strategies achieve perfect recall, enabling confidence variation metrics to be computed for a consistent study population. This tiered evaluation process isolates confidence stability metrics from detection failures.

3.3.2 Detection Robustness

Figure 3.24 summarizes the global detection behavior across the full set of treadmill datasets. A dataset is classified as Recall = 0, displayed in red, if for every shifted position, the target is never detected. The Recall between 0 and 1 orange label is assigned when, within a given dataset, one or more shifted positions are successfully detected, but at least one is missed. The green Recall = 1 label corresponds to perfect detection recall for every translated sample across the dataset sweep. Values are reported as mean across three seeds.

In a test set of 273 datasets, the baseline produced the highest proportion of partial failures (26.1%). Although it does not produce the highest number of complete failures (9.2%), it achieved the lowest proportion of perfect-recall cases (64.7%). This behavior reflects stronger positional bias than the other strategies: the model is often able to detect the target, but only in certain positions, failing to do so consistently across all shifted positions.

BlurPool, which in Section 3.1 yielded subpar results and degraded recall, was able to reduce the proportion of partial failures (21.6%) while achieving a virtually identical proportion of

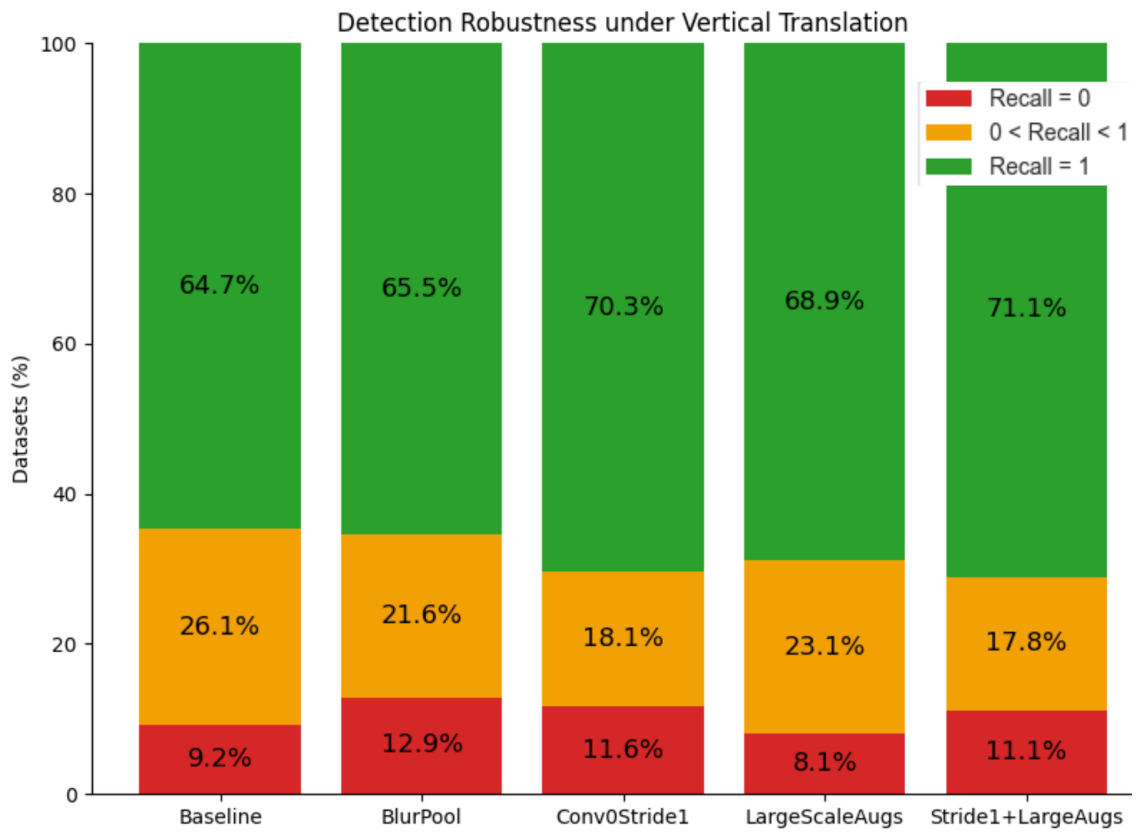


Figure 3.24: Detection Robustness results for tested configurations. Values reported as mean across three seeds. Seed-level standard deviations for complete failure, partial failure, and complete perfect recall are respectively: Baseline ($\pm 0.7\%$, $\pm 2.5\%$, $\pm 2.0\%$), BlurPool ($\pm 1.2\%$, $\pm 2.6\%$, $\pm 1.5\%$), Conv0Stride1 ($\pm 1.7\%$, $\pm 1.9\%$, $\pm 1.3\%$), Large-Scale Augs ($\pm 0.0\%$, $\pm 0.4\%$, $\pm 0.4\%$), Conv0Stride1 + Large-Scale Augs ($\pm 1.5\%$, $\pm 2.7\%$, $\pm 1.7\%$)

perfect-recall cases compared to the baseline (65.5%). However, this improvement comes with a higher number of complete failures (12.9%). This indicates a more polarized behavior: when the model succeeds in detecting the target, it tends to do so more consistently across shifts, but it is also more likely to miss the target entirely in some datasets.

Conv0Stride1 achieved a lower proportion of partial failures (18.1%) and a stronger perfect-recall (70.3%). Despite this, it still has more cases of Recall = 0 than the baseline and data augmentation strategies (11.6%). This suggests that reducing the stride of the first convolution enhances spatial consistency when the object is detectable, but does not inherently improve cases where the object is missed altogether.

The large-scale data augmentation strategy obtained the lowest number of complete failures (8.1%), demonstrating greater robustness against complete detection failure. It also improves the partial failures (23.1%) and complete recall values (68.9%), though not to the same extent as Conv0Stride1. Overall, this suggests that increased data diversity improves detection robustness, particularly by reducing catastrophic failures, while providing moderate gains in positional consistency.

When compared to Conv0Stride1, the Conv0Stride1 + Large-Scale Augs approach achieves moderate gains on all metrics, obtaining the best value for perfect-recall (71.1%) and partial failures (17.8%), though still missing 11.1% of the datasets completely. These results give further evidence that data-focused and architectural strategies work well when employed together.

Summarizing, Conv0Stride1 + Large-Scale Augs provided the best overall performance, improving spatial consistency by minimizing partial failures and maximizing perfect-recall cases. In contrast, Large-Scale Augmentation is strongest at avoiding complete failure cases, ensuring that targets are detected in at least some positions.

3.3.3 Confidence Stability on Common Perfect-Recall Subsets

For an unbiased analysis of confidence stability, it should not be evaluated on datasets where one strategy detects while another fails to do so. This would mix differences in confidence variation with detection robustness, making it unclear whether a strategy is truly mitigating the amplitude of the confidence signal or simply being evaluated on a subset of easier samples.

For this reason, confidence stability was evaluated using per-seed common perfect-recall subsets. For each seed, only datasets where all strategies were able to achieve a recall equal to 1 were retained. This created common subsets that contained 140, 135, and 137 datasets for strategies trained with seeds 42, 41, and 40, respectively, where, within each seed, every strategy detected the target object across the entire vertical shift. The reported results show the average across those three seeds.

This filtering process isolates the effect of the mitigation strategies on confidence variation itself.

Table 3.5 reports three confidence-based metrics. These metrics include the average confidence mean assigned to the matched targets and the confidence standard deviation to assess the overall amplitude of confidence variation across the treadmill sweep. The third metric, $\Delta\text{Conf}/\text{px}$,

Table 3.5: Confidence stability on common per-seed perfect-recall subsets (mean $\pm$ std across three seeds).

Strategy	Conf. Mean $\uparrow$	Conf. Std $\downarrow$	$\Delta\text{Conf}/\text{px}\downarrow$
Baseline (RT)	0.758 ± 0.022	0.038 ± 0.002	0.028 ± 0.002
BlurPool	0.739 ± 0.022	0.032 ± 0.002	0.017 ± 0.000
Conv0Stride1	0.783 ± 0.006	0.027 ± 0.000	0.020 ± 0.000
Large-Scale Augs	0.743 ± 0.017	0.030 ± 0.002	0.019 ± 0.001
Stride1+LargeAugs	0.734 ± 0.004	0.024 ± 0.000	0.017 ± 0.000

measures the mean absolute confidence change between consecutive 1-pixel vertical shifts, making it a direct indicator for local sensitivity to minimal translations.

The baseline exhibited the lowest confidence stability for all tested configurations. It reported the highest standard deviation (0.038), and the largest $\Delta\text{Conf}/\text{px}$ (0.028), indicating greater variability in confidence with respect to placement changes. These values serve as reference points for evaluating the subsequent strategies.

Relative to the baseline, all mitigation strategies achieve stronger results, reducing both the average standard deviation and the local instability of the confidence signal.

Of all the tested strategies, BlurPool yielded one of the lowest $\Delta\text{Conf}/\text{px}$, reducing the mean absolute confidence change per 1-pixel shift from 0.028 to 0.017, displaying the strongest smoothing effect. It also slightly decreases the confidence standard deviation (0.032). Such a result is consistent with the purpose of the BlurPool layer, which is able to reduce abrupt changes caused by small translations by performing anti-aliased downsampling [44]. This should be interpreted alongside the previously analyzed robustness results: BlurPool can improve local confidence smoothness but also produces relatively high values for complete failures.

Conv0Stride1 achieved the lowest confidence standard deviation for single strategies, reducing it to a value of 0.027. Alongside this, it also produced the highest average confidence (0.783), while keeping $\Delta\text{Conf}/\text{px}$ clearly below the baseline (0.020). These results are aligned with those seen in previous experiments, showing that reducing the stride of the first convolutional layer is particularly effective at improving consistency. Across both this and the analysis conducted in Section 3.1, the same architectural change performs strongly without requiring large changes in the training setup.

The large-scale data augmentations strategy also improved confidence stability, though not as much as Conv0Stride1 with respect to confidence standard deviation, and also less strongly than BlurPool in terms of local smoothness. It reduced the $\Delta\text{Conf}/\text{px}$ to 0.019, slightly below Conv0Stride1, and reduces confidence variation to 0.030. These results also reconfirm that data augmentation can reduce confidence variation under spatial shifts, but with more moderate results when compared to architectural interventions.

Conv0Stride1 + Large-Scale Augs reports the smallest standard deviation (0.024) and $\Delta\text{Conf}/\text{px}$ (0.017), though it produces a lower mean confidence value. Once again, these results highlight how

different targeted strategies can integrate harmoniously and achieve even stronger performance.

The comparison with Section 3.1 is also informative because it shows that the same architectural modification yields consistent gains across experimental setups. In the duplicate pair setting, Conv0Stride1 reduced the detection inconsistencies between two visually identical defects appearing at different positions. In the treadmill experiment, the same strategy also improves consistency for a single translated defect across vertical positions. This strengthens the explanation that down-sampling is a key factor in position-dependent instability.

As a whole, these results show that architectural and data-centric approaches can be complementary and help clarify an important distinction between different types of mitigation. BlurPool mainly reduces local, pixel-to-pixel changes, Conv0Stride1 reduces the overall magnitude of the confidence variation, and Large-Scale Data Augmentations provides an intermediate data-driven enhancement. Each successfully mitigates the phenomenon in different ways, but none fully resolves it, preserving the architectural periodicity.

Chapter 4

Conclusions and Future Work

This final chapter consolidates the work conducted throughout this dissertation, revisits the research questions formulated in Chapter 1, explores the implications and limitations of the experimental work, and outlines directions for future research.

4.1 Summary of Contributions

The present work studied position-dependent detection inconsistencies in modern object detection models, specifically translation bias and its effects in a real industrial inspection setting. The work combined a comprehensive review of the wide-ranging literature regarding positional bias, combined with an empirical study grounded on a cork inspection dataset, in which the acquisition process generates visually identical pair defects at a fixed vertical offset. This setting provided a controlled, naturally occurring scenario to analyze how consistent detection predictions are for visually identical translated objects. A shorter version of the present work was submitted as a scientific paper to the British Machine Vision Conference (BMVC).

The literature review explored architectural mechanisms (padding, strided downsampling, positional encodings) and data-related effects (center bias, context bias) as possible sources of positional bias. It also highlighted that most analyses overwhelmingly focus on classification rather than object detection, further motivating a study centered on detector behavior.

The experimental work focused on three major contributions. The development of a custom pair-consistency evaluation pipeline to quantify prediction asymmetries between pair defects. Using this pipeline, significant performance inconsistencies were uncovered: cases when only one of the pair defects was successfully detected, class disagreements, and prediction confidence differences.

Subsequently, a comparative evaluation of mitigation strategies covering architectural, data-centric, and loss-focused approaches was executed. Across tested strategies, reducing the stride of the first convolutional layer from 2 to 1 achieved the greatest single-strategy improvement, particularly for smaller defects, which disproportionately benefited from this approach. The best overall result was obtained by combining it with structure-preserving data oversampling of pair-defect

instances, which underlined the complementarity of architectural and data-centric approaches for resolving positional inaccuracies. The BlurPool approach yielded subpar results in this setting, underscoring that methods developed for classification do not transfer trivially to detection.

To understand the underlying mechanism behind these inconsistencies, an analysis was conducted using a controlled treadmill translation experiment, which uncovered a periodic confidence modulation phenomenon: detector confidence varies systematically under translation according to a periodic pattern. This structure varies linearly with the ratio of the image size to inference resolution. The conducted ablation experiments demonstrate that this phenomenon is attributable to the network’s strided downsampling mechanism and its propagation through the feature pyramid. Removing the stride-32 stage of the detector halves the period, and reducing the stride of the first layer produces the same outcome. A retraining experiment shows how the dominant period is not fixed by the architecture alone. Training distribution influences the stride pathways the model learns to rely on during training, which in turn govern the observed period, explaining why the same architecture produces different periods depending on the training data. This phenomenon was confirmed across YOLOv8 variants with different sizes, across newer YOLO variants (YOLOv11, YOLOv26), and in RT-DETR, a hybrid detection model, reinforcing that this behavior is not specific to a single model. A subsequent confidence-stability study using some of the mitigation strategies previously applied to the pair-defect setting also reduces prediction confidence variability in treadmill datasets, although none eliminate the periodic structure.

4.2 Revisiting the Research Questions

After the experimental work performed, the research questions devised in Chapter 1 can now be addressed.

RQ1 asked to what extent modern object detection models exhibit translation-induced positional bias, and how this bias manifests in a controlled industrial setting. The conducted experiments demonstrate that translation bias affects a variety of **SOTA** detectors which incorporate a **CNN** backbone, not only within the **YOLO** detection family (YOLOv8, YOLOv11, YOLOv26) but also in hybrid models such as RT-DETR. This bias was studied in the cork inspection dataset, where it manifests as detection asymmetries observed between the pair-defects, and quantified using metrics that measured detection consistency, such as missed detections, class disagreement, and confidence variation. In the treadmill translation setting, the bias manifests as the periodic structure analyzed in Section 3.2. This structure was also found on a second dataset, supporting the generality of the findings.

RQ2 asked what architectural and training-related factors govern translation bias in CNN-based detectors, and how these mechanisms can be quantified and visualized. The treadmill translation experiment investigates the causes of the periodic structure. Architectural ablations identify strided downsampling and its propagation through the feature pyramid as the source of the behavior, while sliding-window autocorrelation analysis quantifies the pattern, and accompanying plots support its visualization. The retraining experiment further indicates that the dominant period is

shaped by the stride pathways the model learns to rely on during training, rather than architecture alone. Per-head classification activation maps are used to help visualize this reliance. Together, the developed explanatory tools offer a rigorous way to measure and visualize a phenomenon that traditional global metrics completely fail to capture.

RQ3 inquired what training strategies, data augmentations, or architectural modifications can mitigate translation bias, and how these strategies compare with each other in terms of effectiveness, trade-offs, and complementarity. The carried out studies comparing different approaches found that reducing the stride of the first convolutional layer from 2 to 1 was the most effective single intervention, particularly for smaller defects, although it substantially increased training time. Pair-aware augmentation also yielded improvements and, when combined with the Conv0Stride1 strategy, achieved the best overall results for mitigating detection asymmetries, highlighting the complementarity between the two approaches. Generic large-scale augmentations provided moderate gains, BlurPool reduced local confidence variability, but degraded recall, and custom loss functions did not obtain competitive results.

4.3 Implications

The results obtained throughout this dissertation provide strong evidence that positional and translation bias in object detectors are not primarily caused by data artifacts. Their source is a structural component of CNN’s feature extracting process, shaped by the training distribution, which influences the dominant harmonic period.

The practical implications of this phenomenon are direct: if an object can yield significantly different detections depending on its position, this can lead to inconsistent results. In safety-critical settings, this risk is even more concerning because even if mAP shows strong results for a particular test set, it may still mask position-dependent inconsistencies if targets appear elsewhere.

This limitation highlights the importance of including translation sensitivity metrics and testing in object detector pipelines, particularly when positional robustness is paramount.

4.4 Limitations

Several limitations of the present work should be acknowledged. The experimental study focused on a small subset of detectors, most of them belonging to the same model family. The dissertation prioritized depth of analysis rather than completeness across detector families, and broader coverage would strengthen the generality of the findings.

The literature review highlighted several layers designed to tackle shift invariance or related issues, which could provide valuable insights if integrated into the experimental work. However, many lacked sufficient documentation on how to incorporate them into object detectors, and their implementation complexity placed them outside the scope of this dissertation.

Lastly, the discovery and visualization of a clear periodic structure under translation has not, to the best of the author’s knowledge, been previously presented in the way it is in this dissertation.

Therefore, the understanding of certain observed behaviors remains incomplete (for example, why do some dominant periods appear at a multiple of P_0), and these aspects should be explored in future work.

4.5 Future Work

This work serves as the foundation for multiple directions that can and should be followed. From a theoretical standpoint, a deeper understanding of the confidence variation phenomenon is warranted. In particular, empirical work to better comprehend why some translation sweeps show absolute confidence variation between phases and others don't, as well as an exhaustive analysis regarding the interaction between non-integer resize ratios and harmonic dominance.

Developing shift-invariant and shift-equivariant operators or incorporating existing ones into **SOTA** detection models is also a relevant path to explore so that more researchers can study them. The existing literature is too centered on classification, leaving object detection underinvestigated. Alternatively, creating inference pipelines that are aware of the discovered periodic behavior and are able to apply that knowledge to more consistently detect targets, in an efficient way, is likewise a compelling concept.

A consolidated evaluation protocol that integrates translation sensitivity into standard inference metrics for any dataset is another important step that should be further advanced, as a complement to the currently existing metrics.

Finally, a broader analysis of additional architectures would help establish this phenomenon as a common-knowledge issue that developers must be aware of. Assessing if fully transformer-based detectors also experience similar effects is also a meritorious path to follow.

Concluding, this dissertation characterizes spatial downsampling as a concrete, measurable, and visualizable source of positional bias, and shows how the resulting periodic confidence can have direct consequences for real-world deployments. The proposed evaluation and mitigation pipelines serve as the groundwork for future research aimed at building more robust and trustworthy detection.

References

- [1] Bilal Alsallakh et al. “Mind the Pad—CNNs Can Develop Blind Spots”. In: *arXiv preprint arXiv:2010.02178* (2020).
- [2] Ayoub Benali Amjoud and Mustapha Amrouch. “Object detection using deep learning, CNNs and vision transformers: A review”. In: *IEEE Access* 11 (2023), pp. 35479–35516.
- [3] Aharon Azulay and Yair Weiss. “Why do deep convolutional networks generalize so poorly to small image transformations?” In: *Journal of Machine Learning Research* 20.184 (2019), pp. 1–25.
- [4] Valerio Biscione and Jeffrey S Bowers. “Convolutional neural networks are not invariant to translation, but they can learn to be”. In: *Journal of Machine Learning Research* 22.229 (2021), pp. 1–28.
- [5] Robert-Jan Bruintjes and Jan van Gemert. “Learning to Adapt to Position Bias in Vision Transformer Classifiers”. In: *arXiv preprint arXiv:2505.13137* (2025).
- [6] Anadi Chaman and Ivan Dokmanic. “Truly shift-invariant convolutional neural networks”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, pp. 3773–3783.
- [7] dajiao. *P3 DAGM Dataset*. <https://universe.roboflow.com/dajiao/p3-dagm>. Accessed: 2026-05-20. 2025.
- [8] Jia Deng et al. “Imagenet: A large-scale hierarchical image database”. In: *2009 IEEE conference on computer vision and pattern recognition*. Ieee. 2009, pp. 248–255.
- [9] Peijian Ding et al. “Reviving shift equivariance in vision transformers”. In: *arXiv preprint arXiv:2306.07470* (2023).
- [10] Tausif Diwan, G Anirudh, and Jitendra V Tembhurne. “Object detection using YOLO: challenges, architectural successors, datasets and applications”. In: *multimedia Tools and Applications* 82.6 (2023), pp. 9243–9275.
- [11] Josip Djolonga et al. “On robustness and transferability of convolutional neural networks”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, pp. 16458–16468.
- [12] Alexey Dosovitskiy et al. “An image is worth 16x16 words: Transformers for image recognition at scale”. In: *arXiv preprint arXiv:2010.11929* (2020).
- [13] Maximilian Dreyer et al. “Revealing hidden context bias in segmentation and object detection through concept-specific explanations”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2023, pp. 3829–3839.
- [14] Mark Everingham et al. “The pascal visual object classes (voc) challenge”. In: *International journal of computer vision* 88.2 (2010), pp. 303–338.

- [15] Simone Fabbrizzi et al. “A survey on bias in visual datasets”. In: *Computer Vision and Image Understanding* 223 (2022), p. 103552.
- [16] Xiuquan Hou et al. “Relation detr: Exploring explicit position relation prior for object detection”. In: *European Conference on Computer Vision*. Springer. 2024, pp. 89–105.
- [17] Md Amirul Islam et al. “Global pooling, more than meets the eye: Position information is encoded channel-wise in cnns”. In: *Proceedings of the IEEE/CVF International conference on computer vision*. 2021, pp. 793–801.
- [18] Md Amirul Islam et al. “Position, padding and predictions: A deeper look at position information in cnns”. In: *International Journal of Computer Vision* 132.9 (2024), pp. 3889–3910.
- [19] Joel Janai et al. “Computer vision for autonomous vehicles: Problems, datasets and state of the art”. In: *Foundations and Trends in Computer Graphics and Vision* 12.1-3 (2020), pp. 1–308.
- [20] Glenn Jocher and Jing Qiu. *Ultralytics YOLO11*. Version 11.0.0. Accessed: 2026-05-20. 2024. URL: <https://github.com/ultralytics/ultralytics>.
- [21] Glenn Jocher and Jing Qiu. *Ultralytics YOLO26*. Version 26.0.0. Accessed: 2026-05-20. 2026. URL: <https://github.com/ultralytics/ultralytics>.
- [22] Pratyusha Ria Kalluri et al. “Computer-vision research powers surveillance technology”. In: *Nature* 643.8070 (2025), pp. 73–79.
- [23] Eric Kauderer-Abrams. “Quantifying translation-invariance in convolutional neural networks”. In: *arXiv preprint arXiv:1801.01450* (2017).
- [24] Osman Semih Kayhan and Jan C van Gemert. “On translation invariance in cnns: Convolutional layers can exploit absolute spatial location”. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2020, pp. 14274–14285.
- [25] Alex Krizhevsky, Geoffrey Hinton, et al. “Learning multiple layers of features from tiny images”. In: (2009).
- [26] Jake Lee, Junfeng Yang, and Zhangyang Wang. “What does CNN shift invariance look like? A visualization study”. In: *European Conference on Computer Vision*. Springer. 2020, pp. 196–210.
- [27] Tsung-Yi Lin et al. “Microsoft coco: Common objects in context”. In: *European conference on computer vision*. Springer. 2014, pp. 740–755.
- [28] Wei Liu et al. “Ssd: Single shot multibox detector”. In: *European conference on computer vision*. Springer. 2016, pp. 21–37.
- [29] Pawan Kumar Mall et al. “A comprehensive review of deep neural networks for medical image processing: Recent developments and future opportunities”. In: *Healthcare Analytics* 4 (2023), p. 100216.
- [30] Marco Manfredi and Yu Wang. “Shift equivariance in object detection”. In: *European Conference on Computer Vision*. Springer. 2020, pp. 32–45.
- [31] Hagay Michaeli, Tomer Michaeli, and Daniel Soudry. “Alias-free convnets: Fractional shift invariance via polynomial activations”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2023, pp. 16333–16342.
- [32] Coenraad Mouton, Johannes C Myburgh, and Marelise H Davel. “Stride and translation invariance in CNNs”. In: *Southern African Conference for Artificial Intelligence Research*. Springer. 2020, pp. 267–281.

- [33] Kensuke Mukai and Takao Yamanaka. “Improving translation invariance in convolutional neural networks with peripheral prediction padding”. In: *2023 IEEE International Conference on Image Processing (ICIP)*. IEEE. 2023, pp. 945–949.
- [34] René Ranftl et al. “Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer”. In: *IEEE transactions on pattern analysis and machine intelligence* 44.3 (2020), pp. 1623–1637.
- [35] Joseph Redmon et al. “You only look once: Unified, real-time object detection”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 779–788.
- [36] Renan A Rojas-Gomez et al. “Making vision transformers truly shift-equivariant”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2024, pp. 5568–5577.
- [37] Sourajit Saha and Tejas Gokhale. “Improving shift invariance in convolutional neural networks with translation invariant polyphase sampling”. In: *2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*. IEEE. 2025, pp. 620–629.
- [38] Xu Shen et al. “Patch reordering: A novel way to achieve rotation and translation invariance in convolutional neural networks”. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 31. 1. 2017.
- [39] Krishna Kumar Singh et al. “Don’t judge an object by its context: learning to overcome contextual bias”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, pp. 11070–11078.
- [40] Zikai Sun and Thierry Blu. “Empowering networks with scale and rotation equivariance using a similarity convolution”. In: *arXiv preprint arXiv:2303.00326* (2023).
- [41] Gergely Szabó and András Horváth. “Mitigating the bias of centered objects in common datasets”. In: *2022 26th International Conference on Pattern Recognition (ICPR)*. IEEE. 2022, pp. 4786–4792.
- [42] Antonio Torralba and Alexei A Efros. “Unbiased look at dataset bias”. In: *CVPR 2011*. IEEE. 2011, pp. 1521–1528.
- [43] Kan Wu et al. “Rethinking and improving relative position encoding for vision transformer”. In: *Proceedings of the IEEE/CVF international conference on computer vision*. 2021, pp. 10033–10041.
- [44] Richard Zhang. “Making convolutional networks shift-invariant again”. In: *International conference on machine learning*. PMLR. 2019, pp. 7324–7334.
- [45] Yian Zhao et al. “Detrs beat yolos on real-time object detection”. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2024, pp. 16965–16974.
- [46] Zhaohui Zheng et al. “Zone evaluation: Revealing spatial bias in object detection”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 46.12 (2024), pp. 8636–8651.
- [47] Longfei Zhou, Lin Zhang, and Nicholas Konz. “Computer vision techniques in manufacturing”. In: *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 53.1 (2022), pp. 105–117.