Interpreting Object Detection using B-cos

Martín Bravo
KTH Royal Institute of Technology
Stockholm, Sweden
mebd@kth.se

Julián Alcibar-Zubillaga
KTH Royal Institute of Technology
Stockholm, Sweden
julianaz@kth.se

Abstract

Interpreting deep neural networks remains a critical challenge in advancing AI systems, particularly in tasks such as image classification and object recognition. This project investigates the B-cos method, an inherently interpretable approach for convolutional neural networks and vision transformers. We aim to reproduce the results of the original paper "B-cos Alignment for Inherently Interpretable CNNs and Vision Transformers", evaluate the architectural choices made by the authors, and extend the method to object detection tasks using the YOLO model. Our work involves implementing B-cos layers in ResNet18 for image classification on the CIFAR-10 dataset and adapting the method for object detection using YOLO trained on the Fruit Dataset. Through these experiments, we validate the results of the original paper and provide new insights into the impact of architectural modifications, such as the absence of ReLU and normalization layers. Furthermore, we demonstrate the applicability of the B-cos method to object detection, which was not explored in the original paper.

1 Introduction

With the availability of more data and computing power, deep networks have been able to evolve over the past few years, and we have seen tremendous progress in areas such as image classification [4], emotion analysis [11], and speech understanding [12], and many other fields. Even though these algorithms perform well in many tasks, due to their complexity is very hard to understand them as human beings. They are very difficult to explain and many times uncontrollable [10]. In safety-critical domains such as image classification and object-detection, where decisions can directly impact human lives, such as identifying objects for collision avoidance in self-driving cars [2], this opacity raises ethical concerns and practical risks. As a result, developing methods to make these models more interpretable has become a central area of research, enabling greater transparency, accountability, and trustworthiness in AI systems.

Existing methods such as LIME [8] and SHAP [6] have been widely used to interpret deep neural networks. However, they often rely on post hoc approximations, require large sample sizes, and raise concerns about fidelity to the model's actual decision process. These limitations highlight the need for approaches that provide more direct and efficient interpretability.

The paper "*B-cos Alignment for Inherently Interpretable CNNs and Vision Transformers*" [1] introduces a method for interpreting deep neural networks by replacing the linear transformation in each layer with a cosine similarity-based operation. While effective, the paper leaves some architectural choices unexplored, such as the absence of ReLU, normalization, and bias terms.

This project reproduces the paper's results using ResNet18 on CIFAR-10, critically evaluates the impact of these design choices, and extends the B-cos method to object detection using YOLO on Fruit Dataset. Key contributions include:

- Validation of the original B-cos results.
- Analysis of architectural decisions and their impact.
- Application of B-cos to object detection, demonstrating its broader applicability.

1.1 Code Repository

The code is available at `https://gits-15.sys.kth.se/mebd/b-cos-object-detection`.

2 Related Work

2.1 B-cos

A standard neural network layer calculates $f(x; w) = w^T x + b$, where w is the weight vector, x is the input, and b is the bias. The B-cos method modifies this by replacing the linear transformation with a cosine similarity-based operation, scaling the weight vector to have unit norm:

$$B - cos(x; w) = \hat{w}^T x \cdot |\cos(x, \hat{w})|^{B-1} \tag{1}$$

where $\hat{w} = \frac{w}{\|w\|}$ and B controls non-linearity. This ensures that the output is always positive and bounded by $\|\hat{w}\| \leq 1$ implies $B - cos(x; w) \leq \|x\|$. A larger B suppresses output with low alignment: $B \gg 1$ and $|\cos(x, \hat{w})| < 1 \Rightarrow B - cos(x; w) \ll \|x\|$.

In standard neural networks, the output is $f(x; \theta) = f_L \circ \cdots \circ f_1(x)$, with f_i as the i-th layer. In B-cos, each layer transforms $l_j^*(a_j; W_j) = \hat{W}_j(a_j)a_j$, where $\hat{W}_j(a_j) = |\cos(a_j, W_j)|^{B-1} \circ \hat{W}_j$. The output of the network is $f^*(x; \theta) = \hat{W}_L(a_L) \circ \cdots \circ \hat{W}_1(a_1) \circ a_1$.

A key feature of B-cos is its ability to compute interpretable contribution maps at each layer. For the last layer, the contribution map is **cmap** $= s_L^l(x) = [W_{1\to L}(x)]_L^T \cdot x$, showing how input features contribute to predictions.

This approach embeds interpretability directly into the architecture, providing insight into the network's decision making.

2.2 ResNets

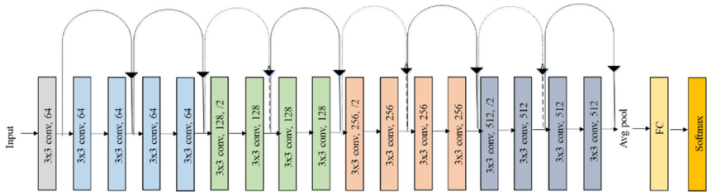

Figure 1: The ResNet18 model consists of 18 layers, including convolutional layers and residual blocks. Residual blocks are the core component of ResNet architectures, and they include skip connections that bypass one or more layers.

Deep neural networks have enabled significant advances in computer vision, but their increasing depth has highlighted critical challenges such as the **degradation problem**. Unlike the **vanishing/exploding gradients problem**, which arises from optimization issues, the degradation problem is structural and leads to higher training errors as networks deepen. Importantly, this problem is not related to overfitting and directly affects training performance.

Residual Neural Networks (ResNet) addressed this challenge by introducing **residual connections**, which reformulate the learning process. Instead of directly approximating a function $H(x)$, ResNet models learn a **residual mapping** $F(x) = H(x) - x$ [5] , where the output will be:

$$H(x) = F(x) + x \tag{2}$$

This approach improves gradient flow and simplifies optimization, allowing effective training of deep networks. Residual links add minimal computational overhead, as short-cut links perform simple element-wise additions. For cases with mismatched dimensions, a linear projection can adjust the link.

Today, ResNet architectures are a **standard benchmark** in computer vision, serving as the basis for evaluating innovations and advances in deep learning. Their robustness and scalability make them essential for testing new ideas in tasks such as image classification, object detection, and segmentation.

2.3 YOLO

You Only Look Once (YOLO) proposes to use an end-to-end neural network that makes bounding box and class probability predictions all at once. It differs from the approach taken by previous object detection algorithms, which reuse classifiers to perform detection.

By taking a fundamentally different approach to object detection, YOLO has achieved state-of-the-art results, outperforming other real-time object detection algorithms by a wide margin [7].

While algorithms like Faster RCNN [3] use a region proposal network to identify possible regions of interest and then perform recognition on those regions separately, YOLO performs all its predictions using a single, fully connected layer.

Methods that use region proposal networks perform multiple iterations on the same image, while YOLO gets by with a single iteration [9].

The YOLO algorithm takes an image as input and then uses a simple deep convolutional neural network to detect objects in the image. The architecture of the CNN model that forms the backbone of YOLO is shown in Figure 2.

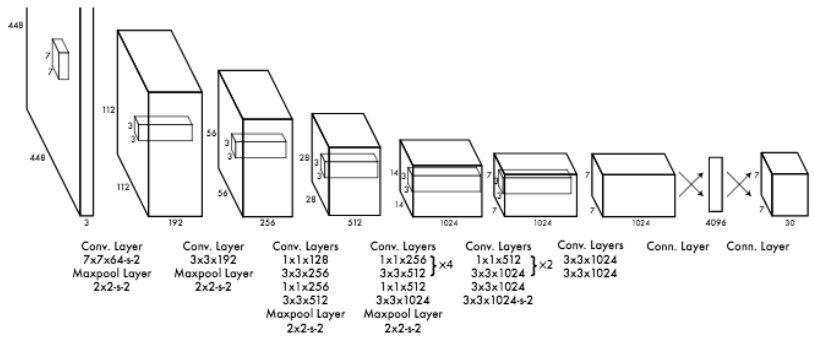

Figure 2: The detection network has 24 convolutional layers followed by 2 fully connected layers. Alternating 1×1 convolutional layers reduce the feature space of the previous layers.

3 Experiments

3.1 Data

CIFAR-10 was used to train the ResNet model, and Fruit Image Dataset was used to train the YOLO model. Though, the results of the B-cos paper were obtained using ImageNet and the original YOLO paper uses Pascal VOC, due to the computational cost of training on ImageNet, we decided to use the selected ones.

3.2 Experiments and Results

3.2.1 Experiment 0: Baseline Validation

To first implement the model we created the B-cos Conv 2d object, then we replaced the convolutional layers of the ResNet with B-cos layers. We set $B = 2$ and did not add any ReLU, MaxOut, or bias. Our first model achieved 84.17% of accuracy after 100 epochs, we got the following interpretability results, a common ResNet18 archives 87% on CIFAR10, this is a good trade-off since we exchange 3% of accuracy but we obtain interpretability.

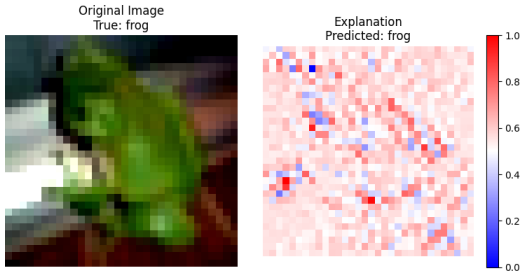

Figure 3: Results obtained using the B-cos model with $B = 2$.

Figure 3 validates that the B-cos layers are achieving their intended purpose of improving interpretability by aligning the model's weights with meaningful input features, while maintaining high classification performance. The focus on task-relevant regions in the explanation map further supports the effectiveness of the B-cos transformation in capturing non-linear and interpretable features.

3.2.2 Experiment 1: Fine-Tuning the Non-linearity Parameter

The B parameter controls the alignment pressure between the weights (w) and the input (x) in the B-Cos transformation. A higher B value emphasizes the cosine similarity between w and x, resulting in larger B values enhance the contributions of strongly aligned features (high cosine similarity) and suppress the influence of weakly aligned features, resulting in a more selective transformation, and model weights are increasingly aligned with task-relevant patterns in the input, producing explanations that highlight features critical for prediction. To investigate the effect of B, we trained neural networks with $B = [1.0, 1.25, 1.5, 1.75, 2.0]$. The results showed improved accuracy and interpretability with increasing B:

B	1.0	1.25	1.5	1.75	2.0
Accuracy (%)	79.37	79.79	80.63	82.34	83.11

This shows that higher B values improve nonlinear feature learning while maintaining good performance, making the model better at capturing task-relevant features. Figure 3 illustrates how explanation maps improve with increasing B. Higher B values result in clearer, more focused maps aligned with the car's features, enhancing both prediction accuracy and interpretability. Lower B values produce scattered, less meaningful explanations.

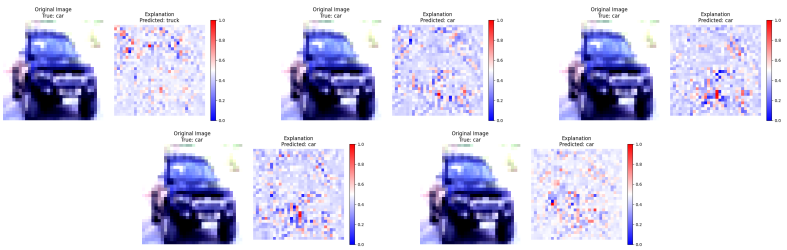

Figure 4: Comparison of explanation maps for different values of B (in ascending order).

3.2.3 Experiment 2: Bias or not?

We explored the effect of enabling bias in the B-cos network. Table 1 shows the accuracy comparison:

Table 1: Accuracy Comparison for Bias Configuration

Bias	Accuracy (%)
False	81.88
True	83.08

Although enabling bias improves accuracy from 81.88% to 83.08%, it reduces interpretability. This is visible in the explanation maps in Figure 5:

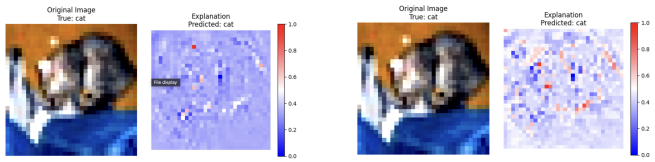

Figure 5: The left image (with bias) appears washed out and less interpretable, whereas the right image (no bias) provides a clearer and more balanced explanation map, highlighting both positive and negative contributions.

3.2.4 Experiment 3: Activation Functions

We compared the effects of ReLU and no activation function on the interpretability of the model. Figure 6 highlights the differences:

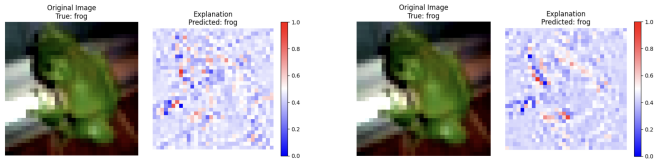

Figure 6: With ReLU (left) produces noisier outputs, while without ReLU (right) yields clearer and more interpretable explanations.

ReLU introduces non-linearity that improves accuracy but reduces interpretability by breaking alignment with input features. Without ReLU, the explanation maps remain clearer and more structured.

3.2.5 Experiment 4: Extending B-cos to YOLO

To evaluate the adaptability of B-cos transformations for object recognition tasks, we extended its framework to YOLO by systematically replacing all convolutional layers in YOLO with B-cos layers. This substitution aims to improve the interpretability of YOLO predictions while maintaining the performance of the original architecture. The implementation directly replaces the linear transformation components of YOLO's convolutional layers with the input-dependent B-cos transformation. As described earlier, the B-cos layers promote weight-input alignment, allowing the model to provide inherently interpretable explanations at each stage of the detection pipeline.

In Figures 7 and 8, we illustrate the explanation maps generated by the B-cos-enabled YOLO for two different examples - a banana and an orange. These examples demonstrate the localized and interpretable nature of the B-cos explanations while preserving the core recognition capabilities of the model.

The methodology can be extended to other object detection models, or even other computer vision tasks, to evaluate how interpretability and performance tradeoffs manifest themselves in different domains.

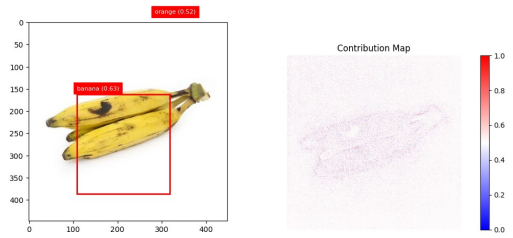

Figure 7: Comparing B-cos YOLO explanation maps and detection results for the banana example.

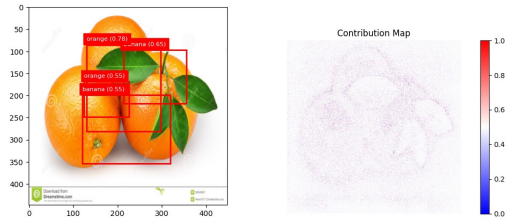

Figure 8: Comparing B-cos YOLO explanation maps and detection results for the orange example.

4 Conclusion

In this work, we reproduced and extended experiments from the original B-cos paper, focusing on interpretability and accuracy trade-offs. Key findings include bias improves accuracy but reduces explanation clarity, ReLU adds non-linearity and boosts accuracy but results in noisier explanations and extending B-cos to YOLO preserves interpretability for object detection tasks.

These experiments highlight that while accuracy and interpretability often trade off, B-cos networks offer a promising approach to balancing the two. Future work includes testing B-cos on additional architectures and datasets to further validate its utility.

5 Extra

5.1 Challenges

As a team, one of the main challenges we faced was the computational power available, which significantly limited the experiments we could run. Specifically, we worked on the project using Google Colab with a T4 GPU. Although we paid for access to this resource, we underestimated the processing capabilities we had acquired, which turned out to be insufficient to run the entire project efficiently. For instance, the final runs of the ResNet experiments with Bcos took days to complete, and at times, Google Colab sessions would disconnect during the experiments. This caused us to restart the experiments from scratch, which caused considerable delays.

Another major challenge was completing the project with only two team members. At the beginning of November, our third teammate withdrew from the course, which complicated the situation and increased our workload. Furthermore, understanding the objective of the paper presented its own challenges. In this regard, the assistant professor was instrumental in helping us ground our approach and structure the project effectively. The paper itself is quite complex, with many nuances that become evident during the experiments, making it difficult to grasp without proper guidance.

We also had to delete a experiment with MaxOut, since implementing it did not improve the accuracy.

5.2 Ethical Consideration, Societal Impact, and UN SDG Alignment

The project aims to provide interpretability to neural networks used for image classification. This is highly valuable in everyday applications, as neural networks are often seen as "black boxes," where the decisions and calculations leading to their performance remain unclear.

Understanding how these networks arrive at their results broadens the scope for making modifications and improvements. One of the areas we worked on, though we could not complete due to limited computational power, was attempting to explain bias. With the knowledge of how Bcos convolutional layers operate, it becomes possible to refine and improve models so that decision-makers can assess whether or not to trust a neural network's output.

This is particularly evident in the YOLO v1 example we worked on. In detection tasks, many factors can drive the decisions and calculations of neural networks. However, when dealing with sensitive matters like health or public safety, humans should prioritize results that make the most sense, taking into account what is activating those neurons.

From a societal perspective, the contributions of this project and the original paper are far-reaching. By making neural networks interpretable, we can increase trust in AI systems and their adoption in critical areas. For example, in our use of YOLO v1 for object detection tasks, understanding the characteristics driving model decisions is particularly important in high-stakes applications such as healthcare diagnostics or identification of safety hazards. Without proper interpretability, there is a significant risk of misuse or over-reliance on AI systems, leading to potential societal harm, such as wrongful decisions or a lack of accountability for errors.

5.3 Individual Contribution.

Martín Bravo took on the role of technical leader and project organizer. He was responsible for designing the overall structure of the project and planning the experiments, ensuring that the objectives were clear and achievable. Martín also played a key role in resolving technical challenges, such as dealing with disconnections and the computational limitations of the Google Colab environment, which were significant hurdles during the project's execution. Additionally, Martín actively contributed to writing sections of the final report.

Julian Alcibar, on the other hand, focused on the experimental development and analysis aspects of the project. He handled the configuration and execution of experiments on Google Colab, utilizing GPUs to maximize efficiency within the computational constraints. Julian was also responsible for implementing the main model based on YOLO v1 and making the necessary adjustments to conduct initial tests. His contributions included analyzing the results of the experiments, identifying relevant patterns related to interpretability, and ensuring the findings were clearly documented. Julian also authored the sections of the report detailing the methodology and the analysis of results.

5.4 Self-Assessment

As a group we believe that we deserve an A. We first reproduced the results of the paper, and discussed its main points and decisions. Secondly, we saw that there were hyperparameters that were badly explained in the paper, and we analyzed them. Finally, we went beyond the scope of the paper and implemented the method for the object detection task, successfully interpreting the predictions of the YOLO model.

There are some bonus points that we believe we deserve:

- There is no implementation of the B-cos YOLO model in the original paper, so we had to implement it from scratch.
- We learned why ReLU, MaxOut and NormLayer were not used in the paper, and we analyzed the effect of using them.
- High quality report.
- Summary of ResNet and YOLO architectures.

References

[1] Moritz Böhle, Navdeeppal Singh, Mario Fritz, and Bernt Schiele. B-cos alignment for inherently interpretable cnns and vision transformers. 2024. URL https://arxiv.org/abs/2406.19407.

[2] Manash Chakraborty and Ahamed Nasif Hossain Aoyon. Implementation of computer vision to detect driver fatigue or drowsiness to reduce the chances of vehicle accident. pages 1–5, 2014.

[3] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. 2014. URL https://arxiv.org/abs/1311.2524.

[4] Robert M Haralick, Karthikeyan Shanmugam, and Its' Hak Dinstein. Textural features for image classification. *IEEE Transactions on systems, man, and cybernetics*, (6):610–621, 1973.

[5] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. pages 770–778, 2016.

[6] Scott M. Lundberg and Su-In Lee. A unified approach to interpreting model predictions. 2017. URL https://arxiv.org/abs/1705.07874.

[7] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. 2016. URL https://arxiv.org/abs/1506.02640.

[8] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. Why should i trust you? explaining the predictions of any classifier. 2016. URL https://arxiv.org/abs/1602.04938.

[9] Ranjan Sapkota, Rizwan Qureshi, Marco Flores Calero, Chetan Badjugar, Upesh Nepal, Alwin Poulose, Peter Zeno, Uday Bhanu Prakash Vaddevolu, Sheheryar Khan, Maged Shoman, Hong Yan, and Manoj Karkee. Yolov10 to its genesis: A decadal and comprehensive review of the you only look once (yolo) series. 2024. URL https://arxiv.org/abs/2406.19407.

[10] HK Xiong, X Gao, SH Li, Y Xu, Y Wang, H Yu, X Liu, and Y Zhang. Interpretable structured multi-modal deep neural network. *Pattern Recognition and Artificial Intelligence*, 31(1):1–11, 2018.

[11] Zhang Ying, Wang Chao, Guo Wenya, and Y Xiaojie. Multi-source emotion tagging for online news comments using bi-directional hierarchical semantic representation model. *J. Comput. Res. Dev*, 55(5):933–944, 2018.

[12] Xin Yu, Yang Jing, Tang Chuheng, and Ge Siqiao. An overlapping semantic community detection algorithm based on local semantic cluster. *Journal of Computer Research and Development*, 52(7):1510–1521, 2015.