NL-FCOS: Improving FCOS through Non-Local Modules for Object Detection

Lukas Pavez
Department of Computer Science
University of Chile, Chile
email: lukas.pavez@ug.uchile.cl

José M. Saavedra
Universidad de los Andes, Chile
email:jmsaavedra@miuandes.cl

Abstract—During the last years, we have seen significant advances in the object detection task, mainly due to the outperforming results of convolutional neural networks. In this vein, anchor-based models have achieved the best results. However, these models require prior information about the aspect and scales of target objects, needing more hyperparameters to fit. In addition, using anchors to fit bounding boxes seems far from how our visual system does the same visual task. Instead, our visual system uses the interactions of different scene parts to semantically identify objects, called perceptual grouping. An object detection methodology closer to the natural model is anchor-free detection, where models like FCOS or Centernet have shown competitive results, but these have not yet exploited the concept of perceptual grouping. Therefore, to increase the effectiveness of anchor-free models keeping the inference time low, we propose to add non-local attention (NL modules) modules to boost the feature map of the underlying backbone. NL modules implement the perceptual grouping mechanism, allowing receptive fields to cooperate in visual representation learning. We show that non-local modules combined with an FCOS head (NL-FCOS) are practical and efficient. Thus, we establish state-of-the-art performance in clothing detection and handwritten amount recognition problems.

I. INTRODUCTION

After the explosion of deep learning [1], we have seen significant advances in different computer vision tasks, not only in research but also in industrial applications. One of the tasks highly favored by these advances is object detection [2]–[6]. So far we know, anchor-based models have shown the highest performance in this task [4]. An anchor is a prior, represented by a starting rectangle that a model tries to transform to fit an object in an image, predicting a tight bounding box. Anchor-based models require extra parameters like the aspect and scales of the anchors to be used, which could be hard to be defined in advance. Furthermore, anchor-based strategies are far from how our visual system does that same task naturally. Indeed, our visual system does not transform rectangles to fit objects in a scene. In contrast, it processes the whole scene taking into account the relationships between different regions to locate objects. This phenomenon is called perceptual grouping and it is well described in the Palmer’s book [7]. The anchor-free detection models are closer to our biological visual system, which intends to predict bounding boxes directly from the whole underlying image. In this domain, models like FCOS [6], CornerNet [8], CenterNet [9] have shown competitive performance without requiring any prior rectangles.

Perceptual grouping studies how the various elements in a complex display are perceived as going together in one’s perceptual experience [7]. This phenomenon was studied by Wertheimer [7] during the early past century, proposing a set of factors called the principles of grouping that includes proximity, similarity, synchrony, among others. The interrelation of different visual components (a.k.a. visual structure) may be learned from experience, particularly during the first months of our lives. A simple example of how different parts of a scene stimulate our visual system is depicted in Figure 1. If we look only at the first row, identifying existing groups doesn’t seem very easy. However, if we examine the second row, we will rapidly note the existence of faces. It would be the consequence of our learned experience.

In computer vision, attention is the mechanism related to perceptual grouping. It has shown to be a critical module in natural language processing [10], and, recently, has also shown a positive impact in computer vision tasks [5], [11], [12].

Since perceptual grouping is critical in our visual system, it should also significantly impact computer vision tasks like object detection, mainly when dealing with anchor-free models. However, these models have not yet leveraged the high interrelation of different receptive fields to define the limits of...
the objects. Therefore, in this work, we evaluate the impact of perceptual grouping through non-local attention mechanisms [10], [13] for anchor-free object detection.

The organization of this document follows the next structure. Section II describes the state of the art in the object detection context. Section III in-depth explains our proposal. Section IV focus on the description of our experiments along with their results. Finally, Section V is devoted to outline the final conclusions.

II. RELATED WORK

In the last years, computer vision has undergone a real boom, becoming a critical area across various industry sectors, including healthcare, agriculture, retail, manufacturing, entertainment, and others.

Object detection is one of the computer vision tasks with multiple industry applications. Its goal is to localize and classify objects in an image or video. The localization is carried out by predicting a bounding box enclosing each target object in an image. For instance, in e-commerce, searching by images is increasingly being demanded. To improve the performance in retrieval, search models decompose a catalog image and a query into semantic components using an object detection tool. 2015 was, notably, a particular year as this marked the expansion of convolutional neural models into a vast number of computer vision tasks like object detection [2], segmentation [14], and synthesis [15]. Faster R-CNN [2] is a milestone in the evolution of object detectors as this was the first end-to-end model using convolutional neural networks, proposing strategies that are still part of the core of current state-of-the-art methods. For instance, training bounding-box regressors guided by anchors and using a region proposal module connected to a shared backbone are some of these core components. An anchor is an a-priori knowledge a model can use to facilitate the learning process and, consequently, the convergence. Formally, an anchor is a rectangle defined by its aspect ratio and scale (area) located at each receptive field in the feature map. In this way, regressors are trained, looking for the parameters that best transform the anchors to fit the objects in the image. Although anchors seem to be a good component for training object detectors, estimating their parameters like scale and aspect could be a challenging task. Moreover, anchor-based training could reduce the generalization to unseen objects. In addition, as we discussed in Section III the use of anchors is far from the natural behavior that, instead, is closer to what researchers call perceptual grouping. This situation has motivated a part of the community to focus on anchor-free models like CornerNet [8], or FCOS [6]. Thus, we will briefly describe the more famous approaches divided by their strategy to learn bounding boxes in the following lines. We have two groups, the first based on anchors and the other with anchor-free models.

A. Anchor-based models

- Faster-RCNN [2]: This is a 2-stage model composed of a region proposal branch called RPN and a classification branch. Both branches share a backbone devoted to learning a visual representation map (feature map). The RPN branch tries to detect objects in a class-agnostic manner, predicting bounding boxes and an objectness score that determines the likelihood of an object. RPN introduces anchors to help it to localize objects. Candidate objects are then passed through the other branch to classify and refine the candidate bounding box with its class information.
- YOLO (from v2): YOLO [3] was presented to provide a real-time model. As its name suggests (you only look once), it comprises only one stage, modeling the classification and object detection by a regression process. The original version was an anchor-free architecture, but due to the success of Faster-RCNN, rapidly, the subsequent versions incorporated an anchor-based strategy.
- RetinaNet [16]: This model incorporates two critical components. The first is a feature pyramid network (FPN [17]) to extract multiscale representations, and the second is the focal loss, a variant of the crossentropy overcoming the bias to the abundant easy regions appearing in an image.
- TridentNet [18]: This is another variant of a multiscale architecture, where the multiple scales are modeled by a variety of receptive field sizes implemented through dilated convolutions (a.k.a. atrous convolutions).
- EfficientDet [4]: It introduces BiFPN and a model scaling strategy. The backbone is the EfficientNet [19] with multiple BiFPN modules stacked as a feature extraction network. As in FPN, each output from the final BiFPN is sent to the classification and regression heads.
- Panet [20]: This extends FPN with a bottom-up augmentation path to allow the low-level semantic features to be propagated to the top layers and semantic high-level to benefit bottom layers. In addition, PaNet incorporates an adaptive feature pooling module.

B. Anchor-free models

- CornerNet [9]: This anchor-free approach looks for an object bounding box by detecting a pair of keypoints, the top-left corner and the bottom-right corner of the target box. Since keypoints lack discriminative visual information, the authors propose four corner pooling modules (top-corner pooling, left-corner pooling, bottom-corner pooling, right-corner pooling) that leverage information on the borders of the target object. In addition, the network produces an embedding vector for each detected corner in such a way that corners of the same object have similar embeddings. A drawback of this model is the high prediction time due to the costly convolutional network used as a backbone (Hourglass-Net).
- CenterNet [2]: This is a natural extension of CornerNet, where the model additionally takes information from the
center of the bounding box to validate the prediction. The authors also propose to merge each pair of corner pooling in a sequential manner, which allows the model to leverage information inside the whole bounding box instead of the borders only.

- **FCOS [6]**: FCOS solves the object detection problem in a per-pixel methodology, where each pixel corresponds to background or is inside an object. FCOS predicts a 4D vector with the distances from the pixel to the border of the target objects (Fig. 2). It uses a ResNet-50 as a backbone, together with a feature pyramid network. A header is connected from each pyramid level with shared weights between the scale levels. The detection heads consist of a classification branch and a regression one. An additional computation is added for centerness at the end of one of the branches, where it shows better results at the regression branch.

As we can see, the predominant backbones in the detection task are based on convolutional neural networks. Their effectiveness for extracting discriminative visual representations makes them also appropriate for other vision tasks. In contrast, attention-based models have been the favorite approaches in natural language processing [10], [21], [22], mainly due to their long-term dependency mechanism. Attention is also a critical component in our natural visual system as it allows receptive fields to interact with each other, producing the perceptual grouping effect.

The combination of attention with convolutional models for computer vision tasks has been studied recently. Some approaches include attention as a complement of convolutional-based backbones [13], [23], [24]. Others have proposed pure attention architectures for visual tasks [8], [11], [25].

The problem with pure-attention architectures is their quadratic complexity, making them impractical to apply at high resolutions. On the other hand, convolutional models have shown to efficiently extract discriminative visual representation in multiple scales [17], [20]. Thus, combining a convolutional backbone for feature extraction with attention modules for perceptual grouping is a natural strategy. Although we have seen outstanding results with anchor-based approaches, the use of anchors adds an extra cost and the difficulty of defining the required parameters. Furthermore, it is far from how our natural system does the same task. Anchor-free models address this problem, but they have not gotten over the performance of anchor-based approaches yet.

Our research indicates that the anchor-free architecture FCOS [6] is effective and efficient concerning others of the same type. In addition, it can be further improved by attention mechanisms. In this vein, the non-local attention mechanism [13] seems to produce the same effect of perceptual grouping as it takes into account the interrelation of different receptive fields to produce a visual representation at a specific point. In addition, the interaction between components is learned during training, simulating the learned experience discussed in Section I.

Therefore, this proposal presents NL-FCOS, an improved version of FCOS, that implements non-local attention mechanism after the feature map computation. Our experiments show new state-of-the-art results for clothing detection as well as for handwritten digit separation (a key component in amount recognition systems [26]).

### III. Non-Local FCOS

We propose a boosted FCOS, called NL-FCOS. This is an FCOS architecture connected to detection heads, which incorporate non-local attention mechanism. Our proposed attention module is based on the Non-Local Neural Networks [13], where it computes a visual representation at a position $i$ as a weighted sum of the features at all positions. This mechanism is also the core of the popular Transformers [10]. Figure 3 depicts a scheme of the non-local attention.

This kind of attention mechanism implements the perceptual grouping occurring in our visual system, where different parts of an image stimulate our system as they were one single stimulus. Since a non-local module applies a function relating each position with the others, it could also learn the identity relationship, which will produce the same result as the original model. Therefore, the proposed NL-FCOS should perform equally or better than the original FCOS.

In Equation 1 we formally define the non-local attention used in this proposal. The output $y_i$ is the resulting visual representation at position $i$, incorporating information from the other positions. The relationship between the representations at $i$ and $j$ is learned during training. This behavior is also connected to our visual system, where the stimuli produced by different components of a scene are learned from experience.

---

**Fig. 2.** 4D regression predicted by FCOS [6].

**Fig. 3.** Scheme of the attention mechanism in a non-local block in NL-FCOS.
In Equation 1, \( \theta, \phi \) and \( g \) are linear transformations that are directly implemented by 3x3 convolutions. Finally, the non-local transformation is then easily computed by a matrix multiplication \[13\]

\[
y_i = \sum_j f(\theta(x_i), \phi(x_j))g(x_j)
\]

(A) NL-FCOS

We choose FCOS as the free-anchor detection architecture because it provides an excellent trade-off between efficiency and effectiveness. As we will see later, FCOS allows us to get competitive precision with a prediction time three times faster than the other anchor-free approaches.

We modify FCOS by adding a non-local module at the beginning of each detection head after the feature extraction stage. This is because the network computes a feature map with the necessary features for the detection. The non-local block will relate each feature with the rest, adding more information to the underlying representation. Figure 4 displays our proposed NL-FCOS architecture. We have also run experiments with multi-head attention, similar to what Transformer does [10].

(B) Shared Non-Local Modules

Similar to FCOS, our proposal NL-FCOS also implements a multi-scale feature extraction process using FPN [17] and PaNet [20]. In a multiscale framework, objects closer to the observer (those appearing bigger in a scene) require bigger receptive fields modeled by deeper layers. While objects far from the observer (those smaller) require smaller receptive fields modeled by the first layers. Through FPN or PaNet, we can model these scale variations accessing the layers that better represent the objects in a scene. In this way, the features produced by the different scales are comparable in visual representation. That is, the multiscale mechanism acts as a normalizer of objects from different scales. Thus, the classifiers and regressors in the different levels of the pyramidal representation should share their learnable parameters.

For the non-local modules, the model needs to fit \( \theta, \phi \), and \( g \). As the visual representations extracted are comparable between the pyramid levels, it is natural that the involved parameters be shared. Therefore, in our proposal, the three transformations \( \theta, \phi \) and \( g \), required by the non-local block, are the same in all the levels.

(C) NL-CornerNet

In the same manner as NL-FCOS, we also extend CornerNet [8] with non-local attention. To this end, we add non-local modules after the feature extraction backbone that, in this case, corresponds to a stacked Hourglass network. Thus, a non-local module is placed at the beginning of the prediction module, replacing the proposed corner-pooling operation. Through the non-local modules, we expect to generalize this pooling operation by adding more information from the full feature map instead of only the sides of each position as CornerNet’s pooling does. Figure 5 illustrates our proposed NL-CornerNet architecture.

(D) NL-Hybrid

We also propose a hybrid network architecture that replaces the NL-FCOS head with the CornerNet’s head. Figure 6 depicts this hybrid architecture.

IV. EXPERIMENTAL EVALUATION

A. Training details

The experiments where run with a 2-GPU server; each GPU corresponds to an Nvidia Titan RTX with 12GB. The hyper-parameter configuration follows the same used in the original proposals. We will further give more details on it.

B. Datasets

We run experiments with two datasets, both related to industrial applications. The first one is Modanet [27], a clothing dataset with around 52000 images containing boxes enclosing objects from 13 different apparel classes. We divide this dataset into training and testing sets. The first one with 42000 images and 214792 boxes. For testing, we have 10000 images...
and 50907 boxes. An example of an image from the Modanet dataset is presented in Figure 7.

The second dataset is related to the handwritten amount recognition task. Recently, Hochulli et al. [26] have shown outstanding effectiveness in amount recognition using a detection-based approach to separate the handwriting digits. Here, we use the two check amount datasets proposed by Diem et al. [28], which were then bounding-box annotated in the work of Hochulli et al. [26]. These are CAR-A and CAR-B, which are described below:

- CAR-A: This consists of 1500 images with 6744 boxes for training; and 509 images with 2328 boxes for testing.
- CAR-B: This consists of 1000 images with 5621 boxes for training; and 500 images with 2812 boxes for testing.

The main difference between these two datasets is the resolution used for capturing the check images. CAR-A images were captured with a higher resolution, producing images with better quality. An example image of each dataset is illustrated in Figure 8 along with their annotated bounding boxes.

C. NL-FCOS

The network is trained with the SGD optimizer for about 90000 iterations having a batch size of 8 images. The learning rate is initially set to 0.01 and is then reduced by a factor of 10 by the iteration 60K. The backbone is previously pretrained on the ImageNet dataset [29].

As shown in Table I we run experiments aiming to decide the branch in the prediction block where the non-local module should be located. We evaluated three branches: classification (NL-FCOS-cl), regression (NL-FCOS-reg) and both (NL-FCOS). Our results show that the non-local block increases the performance regardless the branch where it is located. We see that putting it in one branch or both produces similar results, but they are superior to the original model without attention. Our results also present a new state-of-the-art in the Modanet dataset, where the best reported anchor-based model is TridentNet, achieving an AP of 54.0 [30]. We also run experiments with a multihead attention (FCOS-transformer), running two non-local attention blocks. Here, the results are lower than using only one attention module by almost 1%. It is possible that the model requires more iterations to adjust a greater number of weights generated by the second non-local module that runs parallel to the other.

In regard to inference time, as shown in Table II, the difference caused by adding the non-local module is marginal (this adds 2ms).

We also run experiments with CAR-A and CAR-B datasets. First, we train our model independently in each dataset. We observe that in both datasets, NL-FCOS produces a slight improvement. We also trained our model combining both sets, obtaining higher performance (see Table III). After applying the non-local module, we observe that the mean precision increases to 99.0, which is 0.2% better than the simple FCOS. We also obtain results comparable to the state-of-the-art results obtained by YOLO [31], where we improve these results in the dataset CAR-B.

The handwriting digit detection is a critical component in the amount recognition task [26]. To measure how well our proposal behaves in this context, we add two metrics: accuracy and similarity. The accuracy increments when the predicted amount is the same as expected. The similarity indicates how
TABLE IV
ACCURACY (ACC) AND SIMILARITY (SIM) IN THE AMOUNT RECOGNITION TASK FOR CAR-A AND CAR-B.

| Network  | CAR-A acc | CAR-A sim | CAR-B acc | CAR-B sim | CAR-A + CAR-B acc | CAR-A + CAR-B sim |
|----------|-----------|-----------|-----------|-----------|-------------------|-------------------|
| FCOS     | 48.7      | 89.5      | 72.4      | 95.3      | 82.6              | 96.8              |
| NL-FCOS  | 74.3      | 95.1      | 75.4      | 96.3      | 83.2              | 96.8              |

Fig. 9. Detection comparison between FCOS and NL-FCOS for the Modanet dataset.

Fig. 10. Detection comparison between FCOS and NL-FCOS for the check amount dataset.

close was the predicted amount to the desired amount (for example, the similarity between 2871 and 28071 is 0.89). This helps to identify in which case we are detecting more or fewer digits of the number. The results are shown in Table IV, where FCOS achieves a low accuracy but a high similarity. Here in all cases, using the non-local modules increases the results in both datasets. We can see some results in Figures 9 and 10.

D. NL-CornerNet

The network is trained with Adam optimizer for 250,000 iterations with a batch of 4 images. The initial learning rate is $0.25 \times 10^{-4}$. Due to memory limitations, we reduce the size of the input of the prediction module from $128 \times 128$ to $64 \times 64$ in all our experiments.

Analyzing the results shown in Table V, we note that the non-local module reduces the AP value. This goes contrary to expected and can have different causes. A cause of this behavior could be the Hourglass network that already combines all the features, and doing it again in the prediction module may be damaging to the general result. Another hypothesis is that the network needs more iterations to adapt the weights of non-local modules. We are inclined to the second reason since a non-local module could learn identity attention, keeping the original performance.

A strong limitation of CornerNet is about its inference time (Table VI). This network is much slower than FCOS (almost three times slower), and applying the non-local module adds about 17% of the original network time, which is not that much considering that the network is already slow. Therefore, our proposal NL-FCOS is still a good choice in real-time scenarios.

E. NL-Hybrid

The network is trained with SGD for about 250,000 iterations with a batch size of 8 images. The learning rate is initially set to 0.001, and it is then reduced by a factor of 10 on the iteration 200K. The backbone is pretrained on ImageNet [29].

The experiments with this architecture are divided in two parts. First, we apply the prediction module to only one level of the Feature Pyramid (level P3). The results in these experiments (Table VII) are lower than the results obtained with FCOS and CornerNet alone, but we note again the goodness of using a non-local module as it gives better results than not using it. In the second experiment, we apply multiscale detection using the Feature Pyramid Network. In this case, adding the non-local module does not impact the performance.

V. CONCLUSIONS

We present NL-FCOS, an improved version of FCOS network, augmented with a non-local attention mechanism in the detection head. As shown in our experimental results, the application of attention favorably affects FCOS, increasing the detection precision by 3%, and keeping inference time low. This is not only in the clothing dataset but also in the handwritten amounts dataset. We also explore multhead attention as Transformers, but our results do not show any improvement.

Our experimental datasets are close to industrial applications, Modanet is used in fashion e-commerce, while CAR-A and CAR-B are used in check amount recognition. Our proposal shows better results than the corresponding best reported anchor-based method in both cases.

TABLE V
EXPERIMENTS WITH CORNERNET AND NON-LOCAL MODULE, DATASET MODANET.

| Network       | AP$^{50}$ | AP$^{75}$ | AP$^{90}$ |
|---------------|-----------|-----------|-----------|
| CornerNet     | 67.2      | 82.9      | 73.5      |
| NL-CornerNet  | 61.8      | 79.4      | 68.0      |

TABLE VI
AVERAGE INFERENCE TIME FOR CORNERNET EXPERIMENTS.

| Network       | Time [ms] |
|---------------|-----------|
| CornerNet     | 100.0     |
| NL-CornerNet  | 117.5     |

TABLE VII
RESULTS WITH NL-HYBRID, WHERE (SINGLE) EXPRESS A MONOSCALE DETECTION ON P3, AND MULTI IS A CLASSICAL MULTISCALE DETECTION USING FPN.

| Network        | AP$^{50}$ | AP$^{75}$ | AP$^{90}$ |
|----------------|-----------|-----------|-----------|
| Hybrid (single)| 47.7      | 66.6      | 53.0      |
| NL-Hybrid (single) | 52.2    | 71.2      | 58.2      |
| NL-Hybrid (multi)| 43.3     | 68.4      | 46.4      |
| Hybrid (multi) | 43.3      | 68.2      | 46.5      |
