A Comparative Study on Convolution and Transformer based SOTA Object Detection paradigms and models

Group 6
Abhay Chowdhary, Parthiv Dholaria, Utsav Garg

Problem Statement

Object detection is a fundamental computer vision task that involves identifying and localizing objects within an image. This project aims to compare two state-of-the-art (SOTA) object detection paradigms:

- Convolutional Neural Networks (CNNs) exemplified by EfficientDet
- Transformers exemplified by DETR (DEtection TRansformer)

We will delve into their architectures, strengths, and weaknesses to understand how they approach the object detection challenge and their relative effectiveness

Efficient Det

EfficientDet is a high-performing object detection model based on convolutional neural networks (CNNs). It builds upon the success of Efficient Net, a family of CNN architectures designed for achieving high accuracy while maintaining the computational efficiency.

- **Scales all together:** EfficientDet scales all parts of the model (size, depth) at once for efficient training and inference
- **Better features:** BiFPN combines different levels of detail for richer feature extraction, improving object detection.
- **Handles imbalanced data:** Focal loss tackles the challenge of uneven object categories in datasets, leading to better accuracy.
- **Top performance, less drag:** EfficientDet delivers high accuracy while being faster than previous models, making it ideal for real-time use.

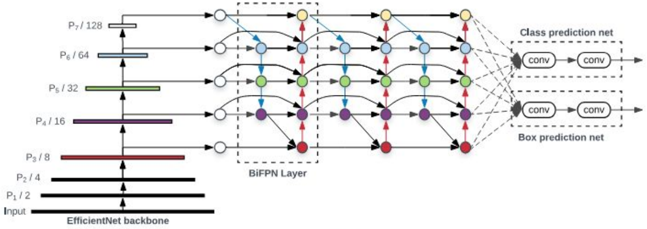


Figure 3: **EfficientDet architecture** – It employs EfficientNet [39] as the backbone network, BiFPN as the feature network, and shared class/box prediction network. Both BiFPN layers and class/box net layers are repeated multiple times based on different resource constraints as shown in Table 1.

Efficient Det

The Backbone

- **Richer Features:** Merges features from various network layers, combining high-level concepts with fine-grained details.
- **Two Pathways:** Employs both bottom-up (low-level to high-level) and top-down (high-level to low-level) information flow.
- **Enhanced Context:** This two-way flow improves the network's understanding of object context within the image.
- **Better Predictions:** Leads to more accurate object detection by providing a more comprehensive feature representation.
- **Efficient Design:** Designed to be lightweight and efficient, making it suitable for use within the EfficientDet model.

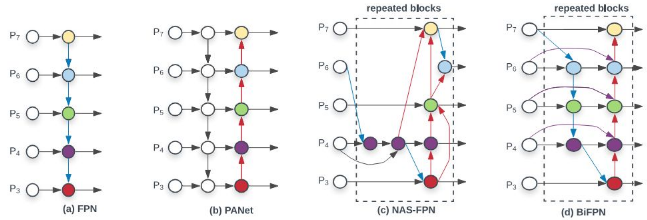
Compound Scaling

Similar to Efficient Net, EfficientDet utilizes a compound scaling approach to create a family of models with varying complexities. This family consists of several predefined EfficientDet models (e.g., EfficientDet-D0, EfficientDet-D7) that offer a trade-off between accuracy and computational efficiency. Users can choose the model that best suits their specific needs, depending on whether they prioritize higher accuracy or faster inference speed

Efficient Det

Bi-directional Feature Pyramid Network (BiFPN)

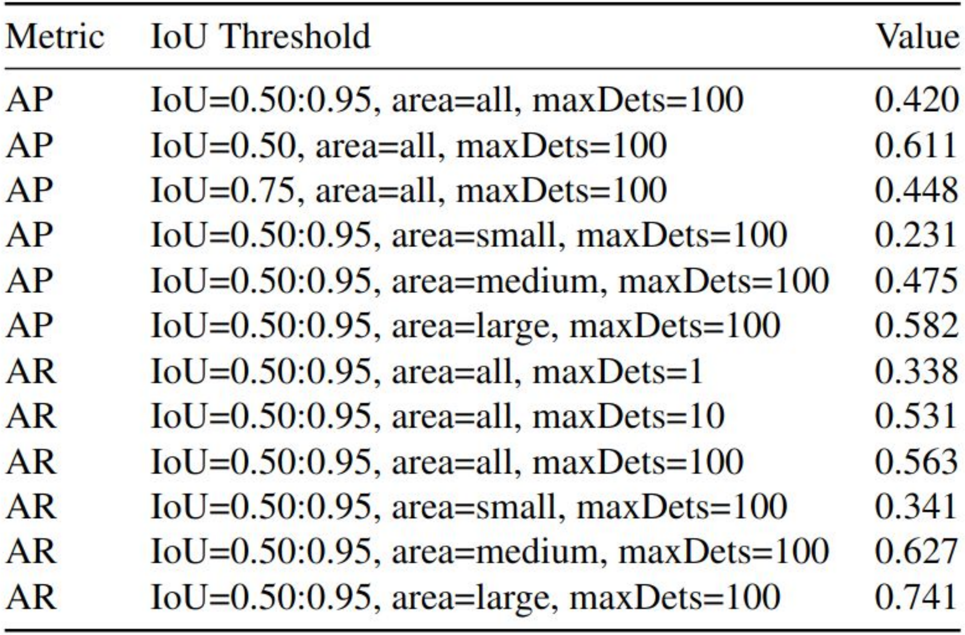
Traditional object detection models often use a Feature Pyramid Network (FPN) to combine features from different network stages. However, FPN is limited in its ability to propagate high-resolution information from shallow layers to deeper layers. BiFPN addresses this limitation of by allowing for bi-directional information flow between different network stages. This enables EfficientDet to capture objects at various scales more effectively, resulting in improved detection accuracy for both small and large object.



(a) FPN (b) PANet (c) NAS-FPN (d) BiFPN

Figure 2: **Feature network design –** (a) FPN [23] introduces a top-down pathway to fuse multi-scale features from level 3 to 7 (P_3 - P_7); (b) PANet [26] adds an additional bottom-up pathway on top of FPN; (c) NAS-FPN [10] use neural architecture search to find an irregular feature network topology and then repeatedly apply the same block; (d) is our BiFPN with better accuracy and efficiency trade-offs.

Inference Result of Efficient Det on COCO

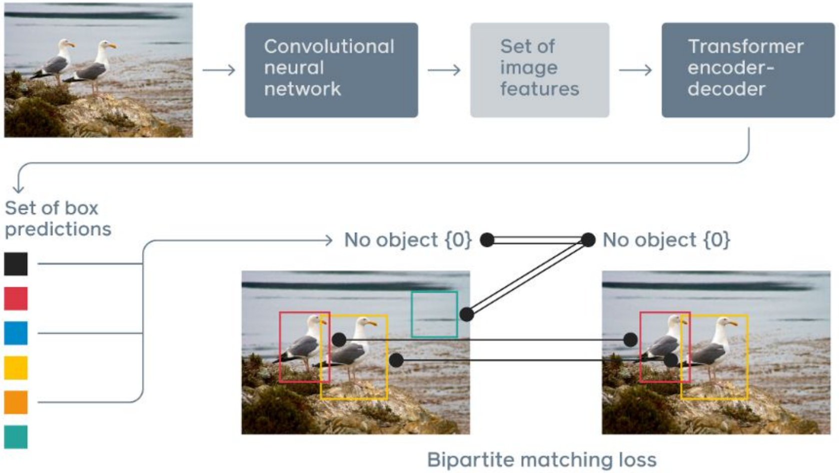


Metric	IoU Threshold	Value
AP	IoU=0.50:0.95, area=all, maxDets=100	0.420
AP	IoU=0.50, area=all, maxDets=100	0.611
AP	IoU=0.75, area=all, maxDets=100	0.448
AP	IoU=0.50:0.95, area=small, maxDets=100	0.231
AP	IoU=0.50:0.95, area=medium, maxDets=100	0.475
AP	IoU=0.50:0.95, area=large, maxDets=100	0.582
AR	IoU=0.50:0.95, area=all, maxDets=1	0.338
AR	IoU=0.50:0.95, area=all, maxDets=10	0.531
AR	IoU=0.50:0.95, area=all, maxDets=100	0.563
AR	IoU=0.50:0.95, area=small, maxDets=100	0.341
AR	IoU=0.50:0.95, area=medium, maxDets=100	0.627
AR	IoU=0.50:0.95, area=large, maxDets=100	0.741

While the official EfficientDet implementation by Google Research is in TensorFlow, we utilized a PyTorch adaptation [Sev20] for this work. This adaptation, as documented on the GitHub repository, removes unnecessary biases in convolutional layers followed by batch normalization, resulting in a slight reduction in model parameters.

DETR

- **Object Detection with Transformers:** DETR uses Transformers, known for natural language processing, for object detection in images.
- **CNN Backbone:** It first utilizes a CNN to extract key features from the image.
- **Transformer Encoder-Decoder:** Then, a Transformer encoder-decoder architecture analyzes the features and relationships between objects.
- **Set-based Prediction:** Unlike traditional methods, DETR directly predicts a set of bounding boxes and object classes.
- **Unique Predictions:** A special loss function ensures each prediction is unique and assigned to one object.

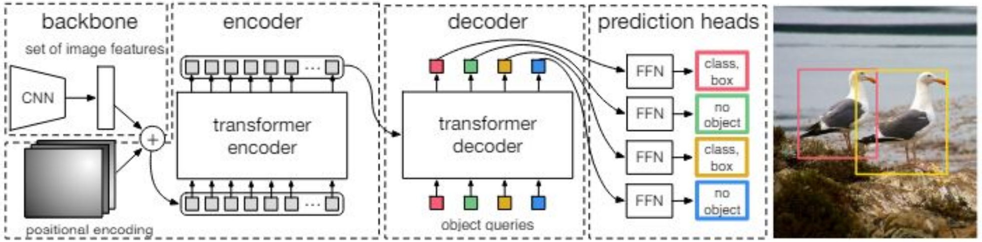


DETR directly predicts (in parallel) the final set of detections by combining a common CNN with a Transformer architecture. During training, bipartite matching uniquely assigns predictions with ground truth boxes. Predictions with no match should yield a "no object" class prediction.

Source: https://ai.meta.com/blog/end-to-end-object-detection-with-transformers/

DETR: The Pipeline

- **Pre-trained Feature Extractor:** The DETR model uses a pre-trained convolutional neural network (CNN) as the backbone feature extractor. This CNN efficiently captures informative features from the input image at a lower resolution.
- **Transformer Encoder:** The CNN features are fed into the Transformer encoder. It uses self-attention to analyze relationships **within** the features.
- **Self-Attention:** Self-attention lets each feature position attend to other positions, identifying informative regions and their connections.
- **Multi-Head Attention:** Multiple attention heads are used in parallel, allowing the model to learn different aspects of the feature relationships.
- **Encoder Output:** The encoder outputs a refined set of features capturing both object details and their spatial relationships.
- **Transformer Decoder:** The decoder also uses self-attention but with a **masking mechanism**. This prevents the decoder from "cheating" by peeking at future outputs during prediction.
- **Decoder Prediction:** The decoder predicts a set of queries, each containing a bounding box and a predicted class.



DETR: The Loss

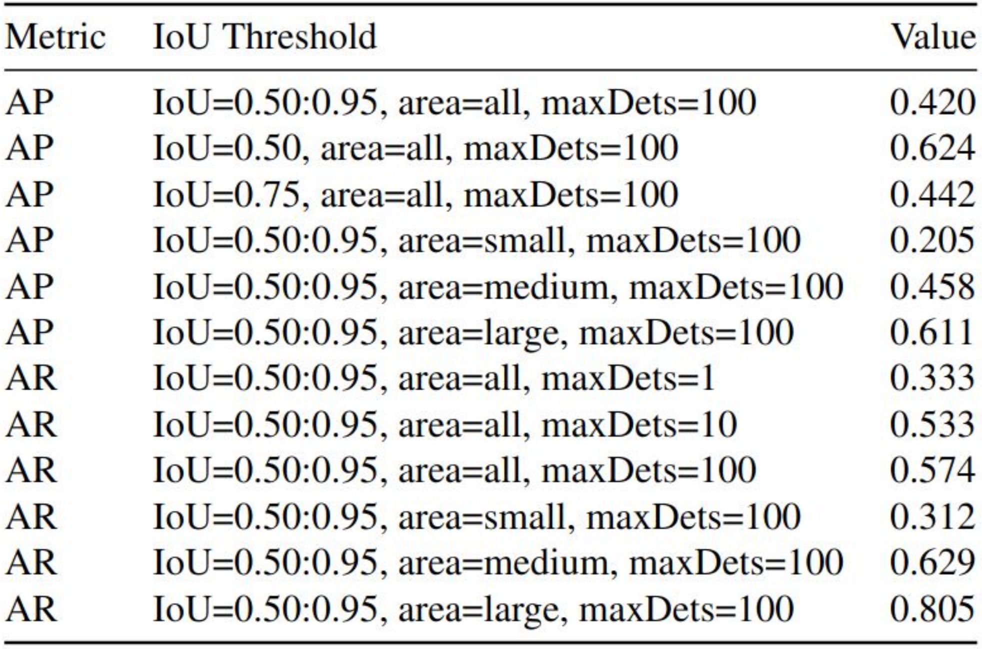
- **Bipartite Matching:** Unlike traditional detectors, DETR doesn't use Non-Maximum Suppression (NMS). Instead, it employs a technique called bipartite matching.
- **Two Sets:** Bipartite matching works by considering two sets: one set containing the predicted bounding boxes and class labels, and the other containing the ground truth boxes and labels.
- **One-to-One Matching:** The goal is to find the optimal one-to-one assignment between these sets.
- **Cost Function:** A cost function (as shown in eqn 1.) is used to evaluate how well each predicted object "matches" each ground truth object.
- **Hungarian Algorithm:** The Hungarian algorithm (as shown in eqn 2.), an efficient optimization method, finds the best possible assignment that minimizes the total cost across all matches. This ensures each prediction is assigned to a unique ground truth object during training.

$$\hat{\sigma} = \underset{\sigma \in \mathfrak{S}_N}{\arg\min} \sum_{i}^{N} \mathcal{L}_{\mathrm{match}}(y_i, \hat{y}_{\sigma(i)}), \tag{1}$$

$$\mathcal{L}_{\mathrm{Hungarian}}(y, \hat{y}) = \sum_{i=1}^{N} \left[-\log \hat{p}_{\hat{\sigma}(i)}(c_i) + \mathbb{1}_{\{c_i \neq \varnothing\}} \mathcal{L}_{\mathrm{box}}(b_i, \hat{b}_{\hat{\sigma}}(i)) \right], \tag{2}$$

Let us denote by y the ground truth set of objects, and $\hat{y} = \{\hat{y}_i\}_{i=1}^{N}$ the set of N predictions. Assuming N is larger than the number of objects in the image, we consider y also as a set of size N padded with $\varnothing$ (no object) and $\mathcal{L}_{\mathrm{match}}(y_i, \hat{y}_{\sigma(i)})$is a pairwise matching cost between ground truth yi and a prediction with index σ(i).

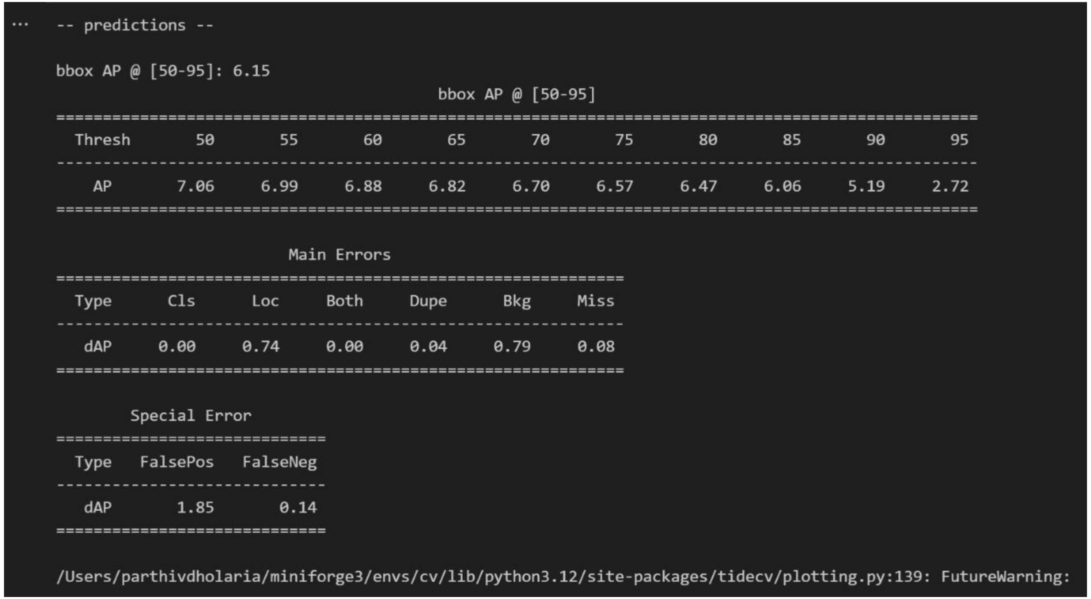
Inference Result of DETR on COCO



Metric	IoU Threshold	Value
AP	IoU=0.50:0.95, area=all, maxDets=100	0.420
AP	IoU=0.50, area=all, maxDets=100	0.624
AP	IoU=0.75, area=all, maxDets=100	0.442
AP	IoU=0.50:0.95, area=small, maxDets=100	0.205
AP	IoU=0.50:0.95, area=medium, maxDets=100	0.458
AP	IoU=0.50:0.95, area=large, maxDets=100	0.611
AR	IoU=0.50:0.95, area=all, maxDets=1	0.333
AR	IoU=0.50:0.95, area=all, maxDets=10	0.533
AR	IoU=0.50:0.95, area=all, maxDets=100	0.574
AR	IoU=0.50:0.95, area=small, maxDets=100	0.312
AR	IoU=0.50:0.95, area=medium, maxDets=100	0.629
AR	IoU=0.50:0.95, area=large, maxDets=100	0.805

We successfully reproduced the exact results reported in the DETR paper ”End-to-End Object Detection with Transformers” by Nicolas Carion et al. (2020).

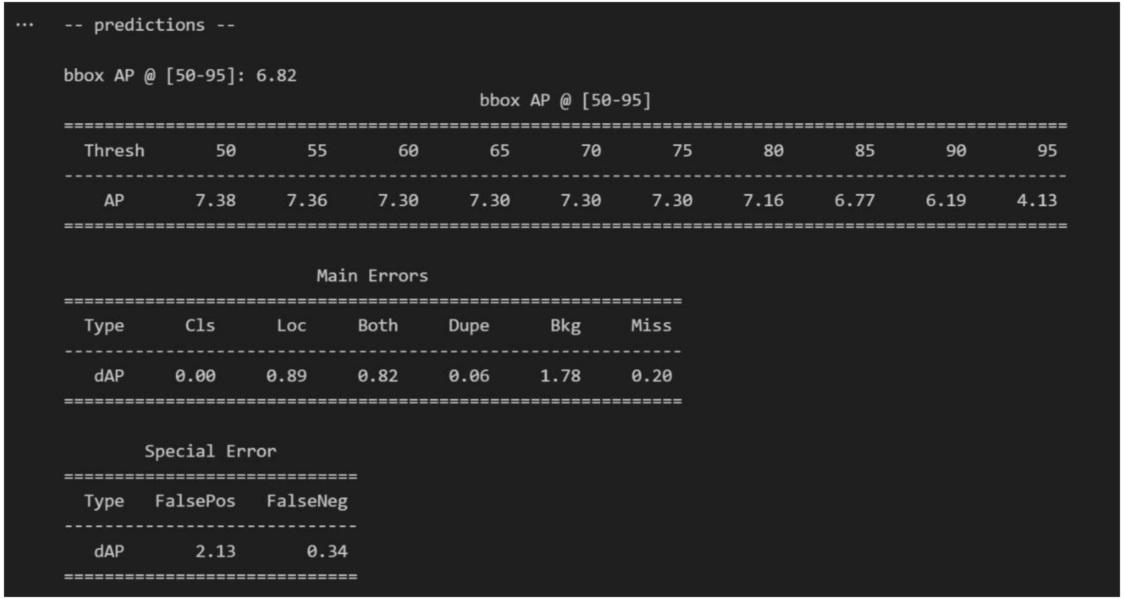
TIDE Analysis of Efficient Det on Pascal VOC 2012



```
...  -- predictions --

     bbox AP @ [50-95]: 6.15
                                              bbox AP @ [50-95]
     ===================================================================================================
       Thresh       50       55       60       65       70       75       80       85       90       95
     ---------------------------------------------------------------------------------------------------
         AP       7.06     6.99     6.88     6.82     6.70     6.57     6.47     6.06     5.19     2.72
     ===================================================================================================

                              Main Errors
     =============================================================
       Type      Cls      Loc     Both     Dupe      Bkg     Miss
     -------------------------------------------------------------
        dAP     0.00     0.74     0.00     0.04     0.79     0.08
     =============================================================

             Special Error
     =============================
       Type   FalsePos   FalseNeg
     -----------------------------
        dAP       1.85       0.14
     =============================

     /Users/parthivdholaria/miniforge3/envs/cv/lib/python3.12/site-packages/tidecv/plotting.py:139: FutureWarning:
```

To evaluate the performance of object detection models, researchers often rely on metrics that consider both the accuracy of the detections and the efficiency of the model. TIDE (Task-oriented Inference for Dense object Detection) is a metric that combines these two aspects. In the context of DETR and EfficientDet models, TIDE can be used to assess how well these models balance accuracy and efficiency for object detection tasks.

TIDE Analysis of DETR on Pascal VOC 2012



```
···  -- predictions --

     bbox AP @ [50-95]: 6.82
                                         bbox AP @ [50-95]
     ==================================================================================================
       Thresh       50        55        60        65        70        75        80        85        90        95
     --------------------------------------------------------------------------------------------------
         AP        7.38      7.36      7.30      7.30      7.30      7.30      7.16      6.77      6.19      4.13
     ==================================================================================================

                           Main Errors
     =============================================================
       Type      Cls       Loc      Both      Dupe       Bkg      Miss
     -------------------------------------------------------------
        dAP     0.00      0.89      0.82      0.06      1.78      0.20
     =============================================================

             Special Error
     =============================
       Type   FalsePos   FalseNeg
     -----------------------------
        dAP       2.13       0.34
     =============================
```

Overall, while DETR shows a slightly higher peak AP, its higher false positives and background errors might inflate its overall detection count, impacting efficiency. EfficientDet, with its lower false positives and background errors, might be a better choice for tasks where precision is crucial.

Qualitative and Quantitative Analysis

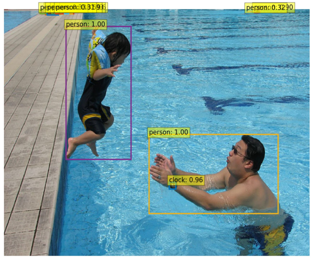
- Calculated the IoU metrics for 20 images in the Pascal VOC dataset for the detection images by both models and results were as follows:
  - DETR: 0.87657
  - EfficientDet: 0.84210

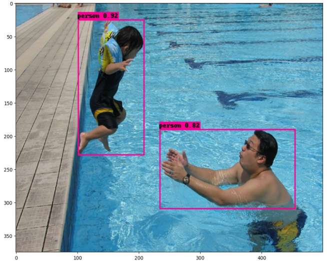
- We observed the following after an in depth analysis of the images comparison:
  - DETR surprisingly outperformed the EfficientDet model for small objects.
  - DETR showed abilities to use surrounding context for object detection.
  - EfficientDet had lower confidence scores even for correct predictions causing a lower threshold to miss detections, while DETR had extremely high scores even for wrong images causing detections which didn't exist.
  - DETR showed very good generalizability by identifying objects found in extremely different and unlikely environments

- Observations made from the TIDE analysis
  - The COCO metric for evaluation (AP @ [50-95]) shows the better overall performance of DETR for the object detection task.
  - Although for the varying IoU thresholding EfficientDet shows consistent performance, while DETR average precision scores drops faster with higher thresholding.
  - DETR shows greater number of false positive (2.13) compared to EfficientDet (1.85), which suggest overall greater incorrect detections potentially impacting both accuracy and efficiency
  - Similarly, the EfficientDet performs slightly better in terms of background errors and missed detections.

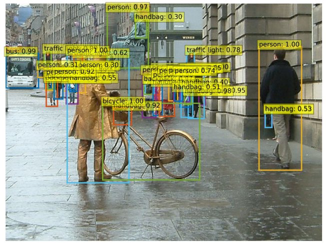
Image Comparison And Analysis

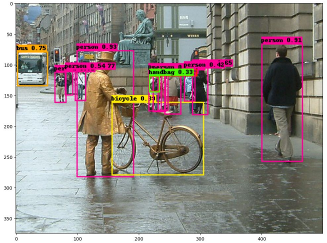
Small object detection





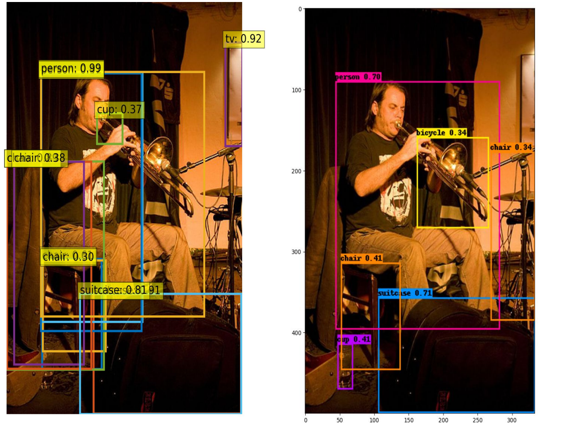
Despite the DETR paper suggesting lower performance on small objects, the presented results are surprising, it successfully identifying a small object (watch) classified as "clock" Outperforming EfficientDet. Also, DETR even detects partially visible people (lower probability) that EfficientDet misses entirely.

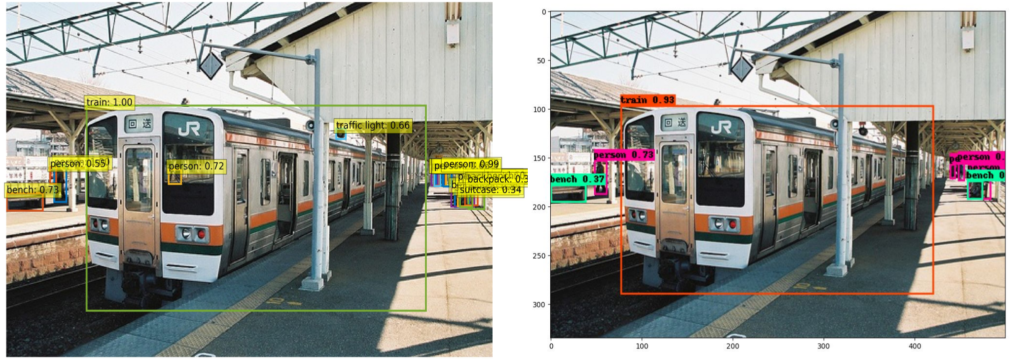




Here the DETR was able to recognize the small handbags while EfficientDet was not able to. Furthermore, DETR also recognized the traffic light which is also barely visible to the human eye.

Context

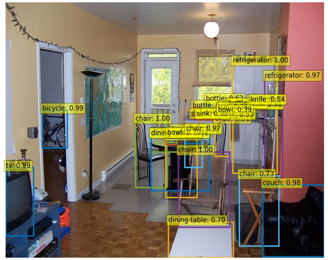


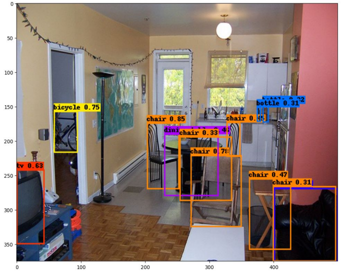


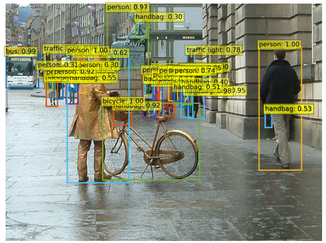
DETR incorrectly identifies the trumpet as a cup, while EfficientDet avoids this mistake since DETR might use surrounding context for detection. The man's hand and trumpet position resemble cup contexts DETR has seen before. This suggests DETR learns not just object form but also surrounding cues (lower confidence implies uncertainty). Subsequently EfficientDet doesn't have a misidentified cup because there's no similar surrounding context for the cup compared to Fig 3. This implies DETR's attention mechanism might play a role in using contextual cues

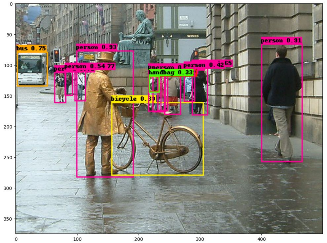
DETR misclassified random unclear image patch inside train position as a person, possibly due to context of seeing person inside trains during training, again an indicator about the use of context for detention decision in DETR because of attention.

Confidence Disparity



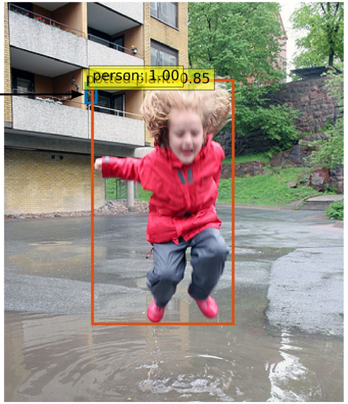


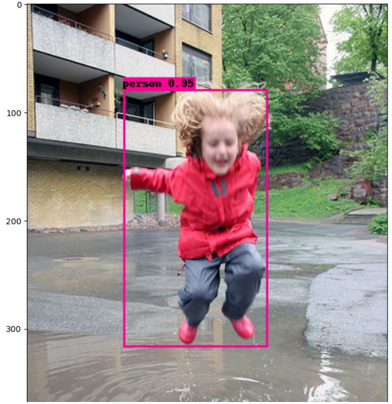




Here we see that even for correct detection of objects, the DETR gives much higher confidence while EfficientDet has much lower average confidence such as (in living room images) bicycle in DETR is with probability 0.99 and in EfiicientDet is 0.72, TV in DETR is 0.99 and in EfficientDet is 0.63 despite both being correct in each case. This average skew causes issues while comparison with same thresholding causing DETR to detect objects not present and EfficientDet to barely miss multiple correct objects

Generalizability





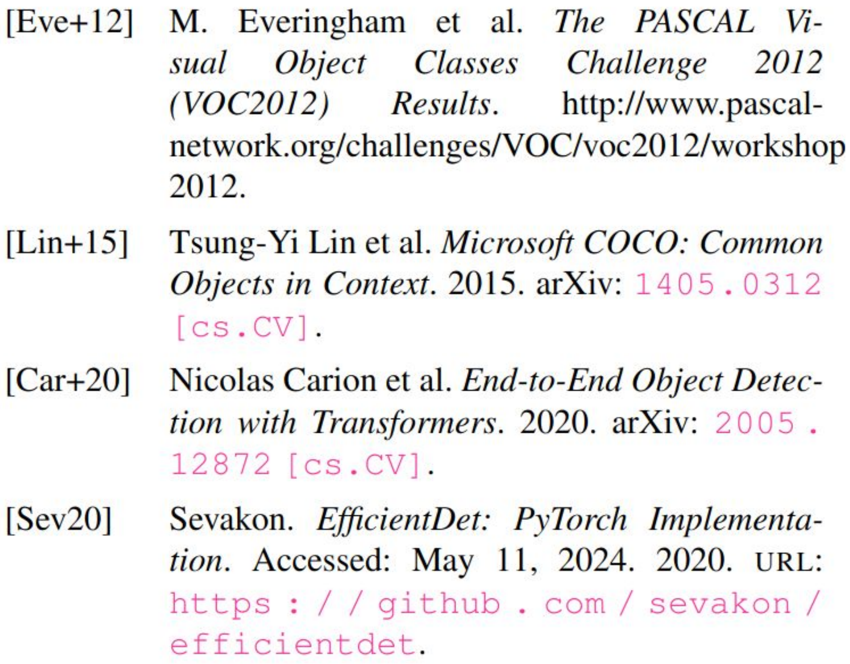
These show how clearly DETR better generalizes to un- seen environments and unlikely setups with its precise identification of the plant (depicted by the arrow) in the building which a human eye would have missed as well, while the efficientDet is unable to do the same.

Individual Contribution

- Parthiv - EfficientDet inference on COCO, Testing on Pascal VOC 2012
- Utsav - DETR inference on COCO, Testing on Pascal VOC 2012
- Abhay - MIoU and TIDE analysis for both the models

We all three together made the slides and the report.

References



[Eve+12] M. Everingham et al. *The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results*. http://www.pascal-network.org/challenges/VOC/voc2012/workshop 2012.

[Lin+15] Tsung-Yi Lin et al. *Microsoft COCO: Common Objects in Context*. 2015. arXiv: 1405.0312 [cs.CV].

[Car+20] Nicolas Carion et al. *End-to-End Object Detection with Transformers*. 2020. arXiv: 2005.12872 [cs.CV].

[Sev20] Sevakon. *EfficientDet: PyTorch Implementation*. Accessed: May 11, 2024. 2020. URL: https://github.com/sevakon/efficientdet.

Thank You