R-VIT

EXPLORING TRANSFORMER-BASED VISION MODELS FOR OBJECT DETECTION

TED STEKETEE

THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF SCIENCE IN DATA SCIENCE & SOCIETY
AT THE SCHOOL OF HUMANITIES AND DIGITAL SCIENCES
OF TILBURG UNIVERSITY

STUDENT NUMBER

2103786

COMMITTEE

dr. Juan Sebastian Olier
dr. Afra Alishahi

LOCATION

Tilburg University
School of Humanities and Digital Sciences
Department of Cognitive Science &
Artificial Intelligence
Tilburg, The Netherlands

DATE

January 15th, 2024

WORD COUNT

8723

R-VIT

EXPLORING TRANSFORMER-BASED VISION MODELS FOR OBJECT DETECTION

TED STEKETEE

Abstract

Object detection plays a crucial role in various applications, including autonomous vehicles, surveillance systems, and augmented reality. Historically, Faster R-CNN demonstrated high precision with its two-stage architecture, and more recently, Vision Transformers (ViT) have shown superior performance in specific computer vision tasks. This study aims to assess the possible benefits of the Region Proposal Network (RPN) for the precision of Faster R-CNN. Furthermore, this study seeks to leverage the potential advantages of incorporating transformers into Faster R-CNN, aiming for a high-precision object detection model. The objective is to compare this model with state-of-the-art counterparts such as YOLOv8 and the original implementation of Faster R-CNN using ResNet-50. Also, a comprehensive error analysis will be conducted. This leads to the formulation of this study's research question, *"How does the performance of R-ViT, which incorporates transformer models in Faster R-CNN, compare to state-of-the-art object detection?"*. The findings reveal that using a Swin Transformer (Shifted Windows Transformer) using a Feature Pyramid Network (FPN) as a backbone improves the model compared to the ResNet-50 variant, particularly excelling in handling small objects and proving most effective for images with numerous objects. However, compared to the current state-of-the-art models, it falls behind on mean Average Precision (mAP). The best-performing Faster R-CNN model that incorporates transformer-based feature extractors is the Faster R-CNN (Swin-FPN) model, achieving a mAP of 0.47.

1 DATA SOURCE, ETHICS, CODE, AND TECHNOLOGY STATEMENT

The Dataset used in this research is the Microsoft Common Objects in Context (MS COCO) dataset. The annotations in the dataset belong to the COCO Consortium and are licensed under a Creative Commons Attribution 4.0 License. The COCO Consortium does not own the copyright of the images. Usage of the images must follow the Flickr Terms of Use.

All the figures in this document belong to the author. Version numbers, libraries, and frameworks are listed in the appendix. Grammarly and PowerThesaurus were used to improve the author's original content for paraphrasing, spell-checking, and grammar. Generative language models GitHub Copilot and ChatGPT were used to debug and improve code. For reference management, Mendeley was used.

2 PROBLEM STATEMENT & RESEARCH GOALS

Object detection is fundamental in various applications, including autonomous vehicles, surveillance systems, and augmented reality. In specific applications, achieving high precision in object detection models is essential to ensure their practical utility and safety. (E.g., Medical Disease Detection, Autonomous driving, and Military Applications). Classical object detection models depend on a two-stage architecture, proposing possible regions of interest and subsequently classifying objects in those regions of interest, leading to high-precision object detection. Also, many classical models utilize Convolutional Neural Networks (CNN) to extract local features from images before passing them through the network. Newer methods omit the proposal stage and aim to predict bounding boxes and object classifications in one pass through the network. (Redmon & Farhadi, 2018; Redmon et al., 2015; Zong et al., 2022) Additionally, novel attention-based techniques, enhance feature extraction in various computer vision tasks. (Dosovitskiy et al., 2020; Z. Liu, Lin, et al., 2021)
The introduction of two-staged object methods initially showcased superior prevision compared to one-stage object detection models, mainly owing to their refined region proposal stage. However, this improvement came at the cost of slower inference. As single-stage object detection models evolved, subsequent advancements ultimately resulted in higher precision models that surpassed the performance of the initial two-stage object detection models.
The current state-of-the-art models have varying approaches to classifying objects. CO-DETR, currently the best-performing model on the COCO-test dev dataset (T. Lin et al., 2014), uses ViTs as the backbone of its network. YOLOv8 is another model that involves numerous convolutional operations and employs grid-based detection. These models do not have a separate region proposal stage.
Given the potential significance of the region proposal stage and the lagging adoption of novel techniques on two-stage models, this research aims to improve two-stage object detection models by implementing novel techniques. Specifically, this research aims to determine the effect of implementing transformer-based feature extractors as a backbone for Faster

R-CNN, a two-stage object detection model, to hypothetically create a high-precision object detection model.
The scientific relevance of this study lies in understanding the importance of the region proposal stage in object detection models to enhance precision and the current role of two-stage object detection models. Also, this study explores how transformers can be integrated with object detection models, examining the benefits, drawbacks, and challenges. Thus, paving the way for more nuanced and refined research into transformer-based models for object detection.
The societal relevance of the project is substantial, with a far-reaching potential across diverse sectors. The enhanced capabilities of object detection models can contribute to more accurate and efficient medical imaging analysis in the medical field. This can lead to improved diagnostics, timely identification of anomalies, and, ultimately, better patient outcomes.
In the realm of self-driving technology, improving object detection models can enhance the safety and reliability of autonomous vehicles. Transformer-based object detection models can provide a more sophisticated understanding of the surrounding environment.
In military applications, a model's ability to analyze complex visual scenes focusing on both local and global features can enhance the detection and recognition of objects of interest. This, in turn, can contribute to improved situational awareness and decision-making in military operations.

2.1 *Research Question*

This study examines the effect of incorporating transformer models in Faster R-CNN, thereby introducing a new model: Region Vision Transformer (R-ViT).

How does the performance of R-ViT, which incorporates transformer models in Faster R-CNN, compare to state-of-the-art object detection?

This study is based on the thesis that the region proposal stage is beneficial for object detection models. Two-stage object detection models will be compared to similar single-stage object detection models on mAP and model complexity to assess the significance of the region proposal stage.

What is the impact of the RPN in object detection models?

Adjusted versions of Faster R-CNN will be constructed that incorporate transformers in their backbone. Multiple implementations of ViTs and

Swin Transformers will be assessed. Also, the effect of adding transformers in the models' RPN and detection head will be evaluated. The models with the best performance will be compared to the state-of-the-art model YOLOv8 onm, mAP, and mAR.

What is the effect of enhancing Faster R-CNN using transformers?

A comprehensive error analysis will be conducted to compare the models. The models will be assessed on over-under prediction, misclassification, and their detection ability on various scales of image object quantity.

What are the most prevalent types of errors in object detection with R-ViT?

3 LITERATURE REVIEW

3.1 *Region Proposal Networks*

Object detection algorithms can be divided into single and two-stage object detectors. Two-stage object detectors consist of two sub-tasks: localizing objects through an RPN and classifying objects at the most promising Region of interest (ROI). Single-stage detectors aim to localize and classify objects in one pass through a neural network. Faster R-CNN is one of the first structures to adopt a distinct region proposal stage. (Ren et al., 2015) Upon its introduction, it surpassed other models on precision. Newer methods that omitted the RPN outperformed Faster R-CNN on speed but not precision. (Z. Liu, Lin, et al., 2021)
The RPN of Faster R-CNN consists of an anchor-generation stage where numerous anchor boxes are suggested. These anchor boxes are passed through a small network that predicts which anchor boxes contain an object. The RPN finally outputs anchor boxes which are most likely to contain objects. These object proposals, combined with the output from the model's backbone, are fed into a compact network to predict the final boxes and classifications.

3.2 *Possible Significance of Faster R-CNN*

An example of a single-stage object detection model is SSD (Single Shot Multibox Detector) (W. Liu et al., 2015). This model employs a single pass through a deep neural network to both localize and classify objects. Similar to Faster R-CNN, SSD utilizes a set of anchor boxes with various aspect ratios and scales for each feature map location.

On a general level, the main difference between Faster R-CNN and SSD is the distinct RPN stage in Faster R-CNN which solely focusses on predicting the most likely object positions before predicting the object classes. Because of this distinction, Faster R-CNN can propose many more anchor boxes in the RPN stage and only uses the best proposals in the final stage, causing the model to be denser in the final stage, compared to SSD.
SSD demonstrates significantly faster processing compared to Faster R-CNN. However, Faster R-CNN can surpass SSD in mAP. Considering the similarity in these two models, these results suggest a significant positive effect for utilizing a distinct RPN in object detection models. (Huang et al., 2016; Kim et al., 2020; M. Li et al., 2020; T. Y. Lin et al., 2017; Tan et al., 2021)

3.3 *Feature Extraction*

3.3.1 *Convolutional Neural Networks*

A broadly used method for feature extraction is the CNN. CNNs use multiple sequential convolutional layers to capture features from images. These layers operate hierarchically, with initial layers focusing on local features, while the higher-up layers capture complex and global features. CNNs manage this by trading spatial dimensions for feature dimensions. This hierarchical structure allows them to extract local features from images effectively.

3.3.2 *Transformers*

ViT, a novel method for feature extraction in images, (Dosovitskiy et al., 2020) applies attention mechanisms to 2D images using transformers. Attention mechanisms allow drawing dependencies between the input and output. (Vaswani et al., 2017)
The main challenge of applying Transformers to 2D images is the computational complexity that scales by a power of 4 for the image size. This problem is addressed by dividing the image into 16 by 16 pixels patches before feeding it into the transformer encoder. A ViT transforms the patches into embeddings that capture the global contextual information for each patch. ViTs make use of a classification token that is added to the input, before passing it through the transformer encoder. This classification token is a learnable parameter that contains all the classification information of the input. In the final stage of the ViT, this classification token is used to classify an image. Due to the use of a separate token to embed all the classification information, the remaining tokens (patches) can contain more spatial information. (Raghu et al., 2021) This differs from CNNs containing

classification information within their spatial tokens.(Deininger et al., 2022) A novel iteration of the ViT is the Swin Transformer. This is a hierarchical Transformer that computes self-attention to non-overlapping local windows and allows for cross-window connections.(Z. Liu, Hu, et al., 2021; Z. Liu, Lin, et al., 2021) This allows for computing self-attention in images using considerably smaller patch sizes. The Swin Transformer adopts the hierarchical characteristic of CNNs by trading spatial dimensions for feature dimensions. This can be a useful advantage when extracting features for object detection.
To cope with the computational complexity of plain Transformers, novel iterations for the Transformer have been introduced. BERT, which is one of the most successful deep learning models in NLP, is a stack of Transformer Encoders used to encode sequential data by applying full attention mechanisms. Because of the full attention used by BERT, it encompasses quadratic complexity. To remedy this, Big Bird was introduced. Big Bird is an implementation of the BERT model that uses sparse attention, which reduces the quadratic complexity into linear complexity. (Devlin et al., 2018; Zaheer et al., 2020)

3.3.3 *Feature Pyramid Networks*

The Feature Pyramid Network (FPN) is an architecture that leverages the inherent hierarchical characteristics of deep convolutional networks to construct feature pyramids at various scales without significant computational overhead. (Vasconcelos et al., 2022) It employs a top-down structure with lateral connections to create high-level semantic feature maps at multiple levels of the network. Implementing FPNs in feature extractors allows for more accurate predictions, with adding minimal computational complexity. Due to the simple architecture of the FPN, its use can be extended beyond the CNN, for example by implementing it with ViTs.

3.4 *Existing Research*

3.4.1 *ViT-FRCNN*

ViT-FRCNN (Beal et al., 2020) attempts to implement ViT as the backbone for Faster R-CNN. To counter the problem that object detection relies heavily on fine details, and ViTs work by slicing the image into patches, this model relies on upscaling the image to a significantly higher resolution. This results in the retention of more refined details. A downside of using higher-resolution images on ViTs is that the number of patches will also increase, which consequently increases the computational cost.
Multiple implementations of ViT-FRCNN are evaluated on the COCO-test

set. These results show no significant improvement in Average Precision for all object sizes except large objects. A subset of ObjectNet, which contains classes that overlap with the COCO dataset is used to test if the model generalizes well beyond the COCO dataset. Because this downscaled dataset consists of almost only large objects, the small objects are omitted. Therefore, the out-of-sample performance is only measured on large objects. This experiment shows that, for large objects, using ViTs as a backbone could improve performance.

3.4.2 *ViTDET*

Another study was performed by Facebook AI Research (FAIR) on using non-hierarchical plain ViT in combination with multiple implementations of Feature Pyramids as a backbone for object detection models.(Y. Li et al., 2022) The ViTs used in this research are pre-trained on ImageNet-1K. The study introduces a Simple Feature Pyramid built from the last feature map of the ViT which enhances the model's overall performance. The final layer of a ViT contains global information. Also, When a ViT is trained on only ImageNet-1K, the lower layers in a ViT cannot learn to attend locally. (Raghu et al., 2021). For object detection, local information is essential for detecting small objects. Considering that the ViTs used in this research are pre-trained using only ImageNet-1K, using a Simple Pyramid Network on only the last layer might not be optimal as a backbone for object detection. The results in this research are expressed in Average Precision. No comment is made on the performance of different sizes of objects. Therefore, no comment can be made on the models' capability to detect small objects using ViTs as a backbone.

3.5 *Sota Models*

State-of-the-art models do not use RPNs. An example of these state-of-the-art models is YOLO (You Only Look Once). This model is constructed of multiple sequential convolutional layers to detect objects. (Redmon et al., 2015) The model has had many updates, the most recent being YOLOv8. Evaluated on the COCO-test-dev dataset, YOLOv8 achieved an Average Precision of 53.9. (Terven & Cordova-Esparza, 2023)
Another object detection approach involves Transformers, with DETR being the first to achieve end-to-end object detection. It utilizes a CNN backbone for feature extraction, a transformer encoder-decoder, and a detection head. (Carion et al., 2020). Recent variations of DETR discard CNNs and adopt ViTs or Swin Transformers as a backbone, relying exclusively on Transformers for object detection. CO-DETR is one such implementation,

introducing a collaborative hybrid assignment training scheme to enhance DETR-based detectors. (Zong et al., 2022) Currently, it achieves the best performance on the COCO-test-dev dataset with a box mAP of 66.0%.

3.6 *Concluding*

While two-stage object detection models have slower training and inference time compared to single-stage object detection models, they demonstrate potential significance in achieving high precision when compared to similar single-stage object detection models. This is the case for Faster R-CNN where its RPN stage, focusing on predicting likely object positions before classifying them, contributes to the precision of Faster R-CNN.
Considering the feature extraction stage, multiple approaches can be used to enhance the Faster R-CNN model. CNNs employ sequential convolutional layers to capture features hierarchically. ViTs offer a novel approach by applying attention mechanisms to 2D images, allowing them to capture global features effectively. The Swin Transformer is an iteration of ViT that computes self-attention in non-overlapping local windows, providing advantages in feature extraction for object detection. Lastly, the BigBird Transformer Encoder introduces sparse attention to cope with the quadratic time complexity of Transformer Encoders, which could present a problem when dealing with 2D images.
The FPN is highlighted as an architecture leveraging the hierarchical characteristics of convolutional networks. It can be used to enhance the backbone by adding minimal computational overhead.
In conclusion, Faster R-CNN shows a possible significance in precision because of its refined RPN stage. Also, novel methods for feature extraction like Transformers and additional FPNs have been shown to enhance object detection models. Where other research focuses on plain ViT for feature extraction, novel iterations have been proposed that could enhance the object detection model and possibly refine the integration of FPNs. Thereby leaving the gap of exploring the integration of novel ViT methods in the Faster R-CNN model to enhance precision.

4 METHODOLOGY & EXPERIMENTAL SETUP

4.1 *Methodology*

4.1.1 *Original Dataset*

The dataset used in this research is the Microsoft Common Objects in Context (MS COCO) Dataset. (T. Lin et al., 2014) The dataset consists of

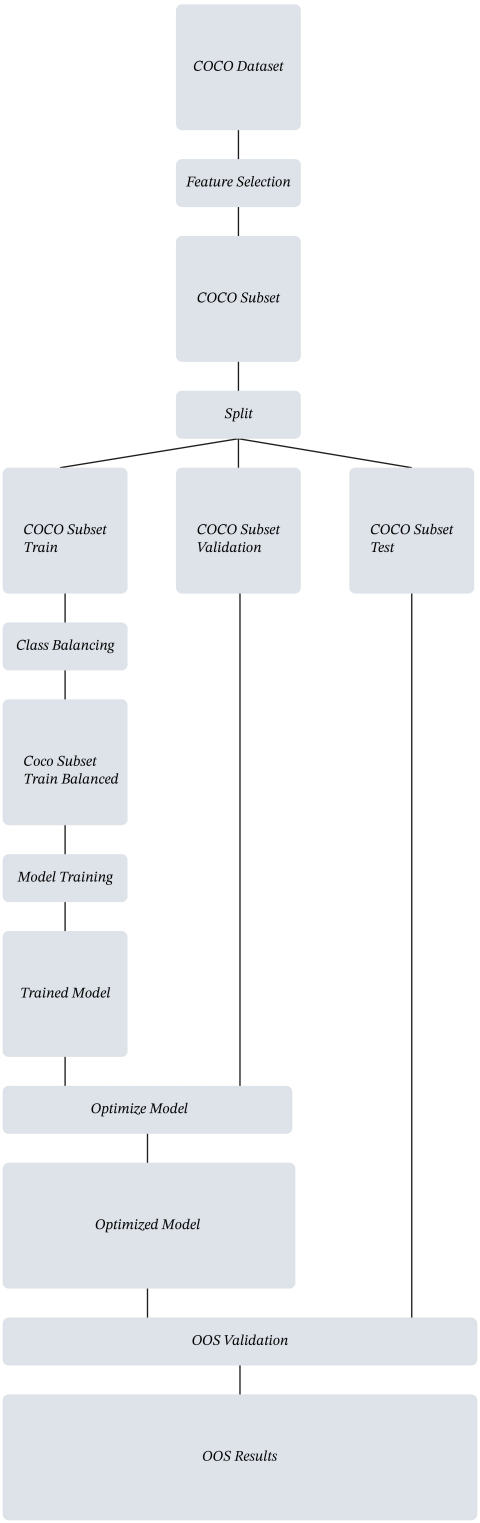


Figure 1: Data Pipeline

over 200.000 common objects with objects categorized in 80 classes divided into 12 supercategories.

4.1.2 *Feature selection*

The selected features from the annotation dataset will consist of bounding boxes and class notations. The dataset will be subsetted to classes relevant to autonomous driving to address computational constraints and reduce training time. The remaining supercategories encompass 'person,' 'vehicle,' 'animal,' and 'outdoor,' which contain 24 classes. To correctly assess the models, all models compared will be trained on this subset of the COCO dataset.

4.1.3 *Exploratory data analysis*

The remaining classes within the subset exhibit significant imbalances, with 'person' and 'car ' being the majority classes. Class balancing is applied to mitigate the risk of overfitting the majority classes. Since sufficient data exists in the dataset, undersampling will be used on the majority classes of the training dataset. To correctly undersample the dataset, annotations will be systematically removed on a per-image basis to prevent images from having deficient annotations. Images with multiple classes containing over 40% in the class 'person' and over 45% in the class 'car' get omitted. After balancing, the training set consists of 30352 images with 114257 corresponding annotations.

4.1.4 *Split*

The original dataset contain a training and validation set. The training dataset is subdivided into training and testing datasets. After the split, the final dataset sizes are shown in Table 1

Set	**Size**
Validation Annotations	19072
Validation Images	3904
Train Annotations	101448
Train Images	27286
Test Annotations	45356
Test Images	9217

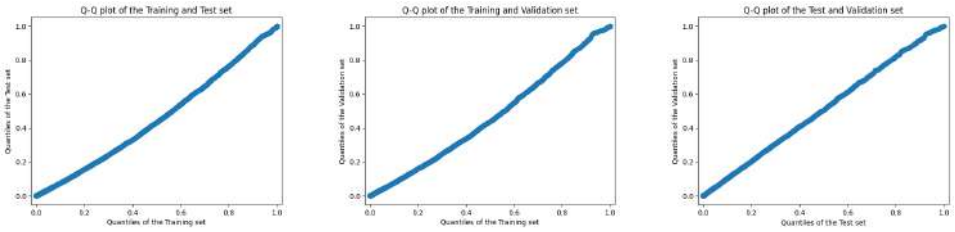
Table 1: Dataset Sizes

4.1.5 *Object size distribution*

In object detection, the size of objects in an image plays a crucial role. Smaller objects pose a greater challenge for object detectors, leading to localization and classification difficulties. Statistical tests are performed to compare statistics of object size, proportional to its image size, in the training, test, and validation set. The Welch's t-test and Mood's median test are used to compare the means and median accordingly for the three data subsets. Also, Q-Q plots are derived and interpreted to compare the distributions.
Results of the Welch's t-test show that the proportional object size of the training and validation set have equal means $t(28140) = 10.7, p < .001$ and that the proportional object's size of the training and the test set have equal means $t(95471) = 17.0, p < 0.001$.

Results of the Mood's mean test show that the median of the training, validation, and test set are equal $\chi^2(2, N = 165876) = 185.6, p < 0.001$.

The Q-Q plots in Figure 1 show similar results.



(a) Q-Q plot Training Test (b) Q-Q plot Validation Test (c) Q-Q plot Test Validation

Figure 2: Q-Q plots for the subsets

4.2 *Experimental Procedure*

4.2.1 *Training/validation loop*

Images and their corresponding annotations are randomly selected in batches of 32 images. The images are transformed into tensors and bilinearly resized to a specific height and width. Experiments are performed using different image sizes comprising 224^2, 512^2 for plain ViTs, and 800^2 for Swin Transformers.
The training loop consists of passing each batch of images through the model. For the Faster R-CNN models, four losses are calculated which consists of two losses for both the RPN and the detection head. The RPN losses consist of the Objectness loss, which measures the probability that an object exists in a proposed RoI, and the bounding box regression loss, which gives the error between the proposed RoI and ground truth boxes.

The detection head losses consist of a classification loss, which measures the classification performance of the detection head, and a bounding box regression loss which gives the error between the predicted bounding boxes and the ground truth bounding boxes. The RPN and the Fast R-CNN prediction head are being jointly trained on the weighted sum of these losses. The optimizer used to train the models is AdamW. (Loshchilov & Hutter, 2017)

4.2.2 *Metrics*

Average Precision (AP), Mean Average Precision (mAP), and Mean Average Recall (mAR) are used to evaluate the models' detection capabilities. The AP will be evaluated for an Intersection Over Union $(IoU) \in [0.5, 0.75]$. The overall mAP and mAR will be computed to compare the models on a single scalar. Both mAP and mAR will be computed on three different levels of object sizes. These levels divide the objects into small, medium, and large objects with bounding box sizes of $[0^2, 32^2), [32^2, 96^2), [96^2, \infty)$ accordingly. Computational complexity will be measured in Floating-point Operations (FLOPs).

4.3 *Experimental setup*

4.3.1 *RPN Significance Testing*

Prior research suggests that models that have a distinct region proposal stage can achieve higher precision. Two experiments will be conducted to assess this.
The performance of a modified version of Faster R-CNN that omits the RPN will be compared to the original implementation of Faster R-CNN. The RPN consists of anchor generation, a stack of convolutional layers, and two siblings fully connected layers to predict what anchor boxes are most probable to contain an object. The RPN will pass on the n best proposals to the detection head. In the modified version of the Faster R-CNN model, all the layers comprising the RPN except the anchor generation will be omitted. This causes the model to pass on all the anchor proposals as object proposals. To limit the amount of object proposals, the anchor size will be set to 64 with a ratio of 1.
To assess the effect of the RPN in Faster R-CNN, a comparison will be made between Faster R-CNN and SSD. The selection of SSD for comparison is based on the similarity between the two models, excluding the RPN.
Also, a comparison will be made between for Faster R-CNN using a Swin-FPN backbone with a modified version of this model that omits the RPN.
For all the models, a fixed image input size of 800^2. For all the experiments,

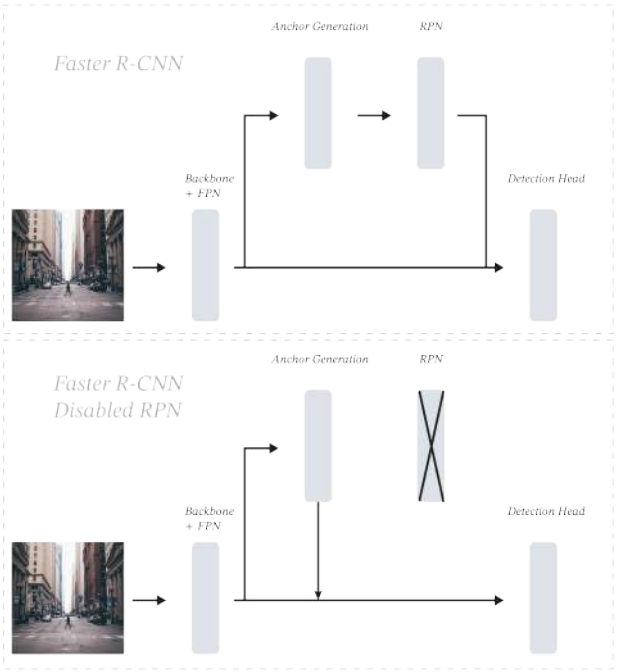


Figure 3: (Disabled) RPN Significance Test

the mAP and computational complexity will be used to compare the models.

4.3.2 *Model construction*

Vision Transformer Backbone
ViTs divide an image into patches and pass these through an encoder. It adds a class token and transforms all tokens into embeddings based on attention mechanisms. Because a ViT's main utility is image classification, it utilizes only the class embeddings outputted by the transformer and omits the remaining tokens. The purpose of these omitted tokens is to serve as features for the class token. Because the remaining tokens embed information about the image, they can be used to construct a feature map that encapsulates the features of the image.
In this approach, the tokens outputted by the Transformer are used to reconstruct a feature map, serving as a backbone for the Faster R-CNN model. Because the class token can encapsulate information about the class of the image, more positional information is preserved in the output of the positional embeddings. (Raghu et al., 2021) Therefore, the class token is preserved while transforming the positional embeddings. When the

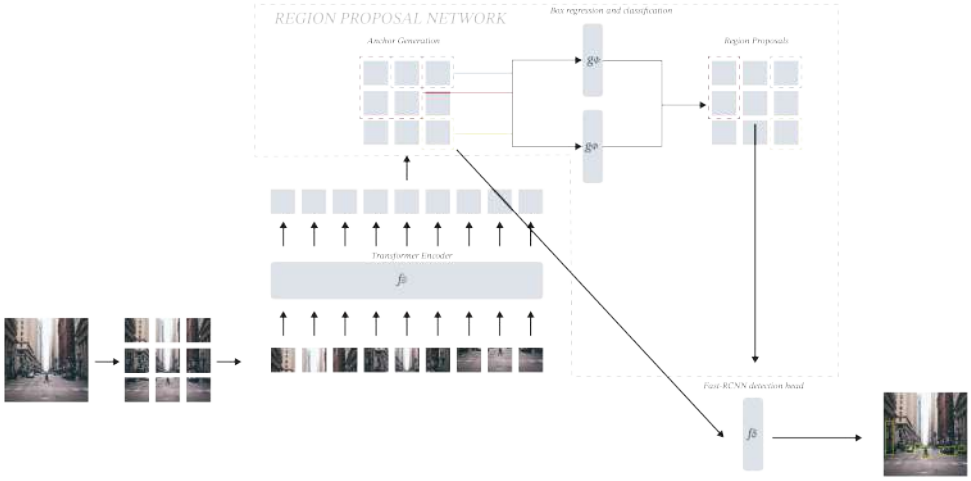


Figure 4: Faster R-CNN (ViT)

embeddings are obtained, the class embedding will be omitted.
Experiments will be conducted using ViT-b/16, ViT-l/16, and ViT-h/14.

Swin transformer backbone
A similar approach will be used when using a Swin Transformer as a backbone. The Swin Transformer used in this research is the Pytorch implementation of Swin Transformer V2. (Z. Liu, Hu, et al., 2021; Paszke et al., 2019) This implementation uses res-post-norm, which moves the LayerNorm of a Swin Transformer Block to the end of the block, making it easier to scale.

Feature Pyramid Network
Experiments are conducted using a FPN to extract feature maps at multiple scales and stages in the backbone. Since a Swin Transformer is inherently hierarchical, the earlier stages of the Swin Transformer contain features of local features, while the later stages contain global features. Combining these feature maps can be useful in detecting objects.

4.3.3 *Hyperparameter Tuning Faster R-CNN*

In this section, the results of the hyperparameter tuning process will be thoroughly examined and discussed. The hyperparameters discussed in this section optimize the backbones' compatibility with the Faster R-CNN architecture.
ViT-b/16 is used as a backbone for the first tuning of the anchor sizes and

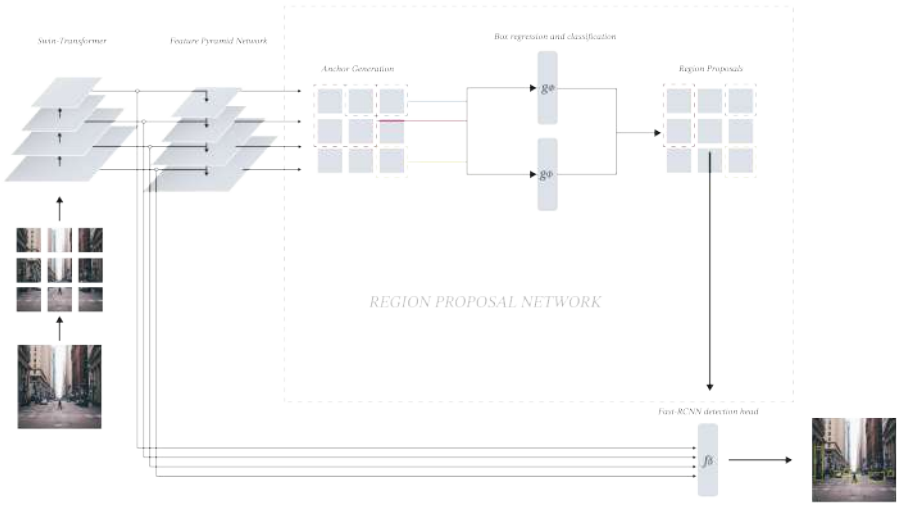


Figure 5: Faster R-CNN (Swin-FPN)

RPN convolutional depth. Results from these experiments will be used in further sections when comparing other implementations of the backbones.

Hyperparameters: Anchor-Sizes
Because the original Faster R-CNN implementation can handle varying input sizes, anchor box sizes representing small and large area sizes on multiple scales are selected. Since ViT-b/16 has a fixed input size of 224^2, the output size will be fixed at 14^2. This fixed output size can be leveraged to select anchor box sizes that represent small, medium, and large areas. The original Faster R-CNN implementation uses anchor boxes of size $32^2, 64^2, 128^2, 256^2, 512^2$. These anchor box sizes will be compared to anchor box sizes of size $2^2, 4^2, 8^2, 12^2, 14^2$.

Hyperparameters: RPN Convolutional Depth
The original Faster R-CNN model initially proposed a single convolutional layer for its architecture. (Ren et al., 2015) However, recognizing the potential impact of varying convolutional depths on performance, experiments are conducted with different configurations, specifically exploring depths of 1, 2, and 3 layers.

4.3.4 *Hyperparameter Tuning Backbone*

Fine-tuning depth
The ViTs used as a backbone are pre-trained on imageNet-1K. Research shows that fine-tuning the backbone for object detection tasks can positively

affect the model's generalization capabilities. Other research contradicts this and shows that freezing the complete backbone can be beneficial for the model. (Vasconcelos et al., 2022) Experiments are conducted on fine-tuning the backbone on varying depths of 3, 6 and 12 trainable backbone layer.

Backbone Size
Changing the ViT size can significantly impact how features are extracted from the image. (Dosovitskiy et al., 2020) proposes a base (ViT-b/16), a large (ViT-l/16), and a huge (ViT-h/14) model. This experiment examines the impact of increasing the ViT backbone from a ViT-b/16 to a ViT-l/16 and a ViT-h/14. For this experiment, the settings for the Faster R-CNN model are maintained identically.

Plain Swin Transformer
Swin Transformers allow for varying input sizes. Increasing the image size can lead to better precision but will also require more computational power. To achieve a compromise between precision and computational efficiency, a fixed input size of 800^2 pixels is used. The RPN and Fast R-CNN detection head settings are derived from the optimization procedure in section 1. An input of 800^2 pixels will generate an output feature map of 13^2. Anchor box sizes will be set to $[2, 4, 6, 8, 12]$.

Swin FPN
FPNs can be used to exploit the inherently hierarchical characteristics of a network. (T.-Y. Lin et al., 2016) Since this is the case for Swin Transformers, this experiment adds a FPN on top of the Swin Transformer. The FPN utilizes four layers as input, sourced from the output of each stage of the network.

Swin Size
This comparative analysis involves three variants of the Swin models: Swin-t, Swin-s, and Swin-b, as outlined in the Swin paper (Z. Liu, Lin, et al., 2021). The methodology adopted includes the incorporation of the FPN, as detailed in the Swin-FPN section, and the implementation of Faster R-CNN, as specified in the Faster R-CNN section.

4.3.5 *Additional Transformers and Optimizations*

Loss function
Faster R-CNN employs binary cross-entropy for RPNs objectness loss and cross-entropy for the detection head classification loss. Since the proposal

of Faster R-CNN, novel, superior loss functions have been introduced. Focal loss (T. Y. Lin et al., 2017), used in RetinaNet, addresses the foreground-background class imbalance problem that is inherent in object detection models. Experiments will be conducted by implementing focal loss in the Faster R-CNN model, replacing the original binary cross-entropy loss and cross-entropy loss.

Transformer RPN
The RPN of Faster R-CNN consists of an anchor generation layer, an intermediate layer consisting of n convolutional layers, and two sibling fully connected layers. The intermediate convolutional layer is used to map the input features into the fully connected layers. Since the new model uses a transformer-based backbone, replacing the convolutional layer with a transformer encoder layer could be beneficial. Experiments will be conducted on replacing the intermediate layer of the RPN for transformer encoder layers. Sparse attention mechanisms proposed by (Zaheer et al., 2020) will be used to cope with the large input size of the backbone and the quadratic complexity of the transformer encoder.

Transformer Detection Head
The Fast R-CNN detection head contains a stack of n convolutional layers. Experiments will be conducted on replacing the convolutional layers of the detection head with a transformer encoder comprised of n layers.

4.4 *Model comparison*

The best-performing implementation of the Faster R-CNN using Swin as a backbone, the best-performing implementation of Faster R-CNN using ViT as a backbone, and the Faster R-CNN using Resnet-50 as a backbone will be compared using the mAP metrics. These results will be compared to YOLOv8.

4.5 *Error analysis*

In this error analysis, confusion matrices of all models are compared to identify prevalent errors. The confusion matrices are generated using an IoU > 0.7 threshold. The confidence score for the Faster R-CNN models is set at 0.6. Confusion matrices are row-normalized for interpretability.

4.5.1 *Over-under predictions*

Optimally, object detection models precisely predict the number of targets in an image. However, most models predict too many (over-prediction) or too few (under-prediction) objects in an image. To assess the different models' detection capabilities, over- and underpredictions are compared for all models.

4.5.2 *Misclassification per class*

An assessment of misclassifications per class is conducted to identify the most common misclassifications. This involves analyzing the confusion matrix, excluding the correct, over, and underpredictions.

4.5.3 *Number of objects comparison*

The average precision of an image is expected to be affected by the number of objects present in it. Images containing fewer objects are expected to exhibit a higher average precision, whereas the average precision decreases with an increasing number of objects. The objective of this comparison is to assess the models' detection capabilities on multiple number of objects per image.

5 RESULTS

In this section, the results of the experiments mentioned in the Experimental Setup will be discussed. For the evaluation of the models, the mAP and the losses will be interpreted. The interpretation of the losses will be mainly used to monitor the training progress and to detect if the model is under or overfitting on the training data. The mAP will be used to evaluate the model's generalization capabilities.

5.1 *RPN Significance Testing*

5.1.1 *Disabled RPN*

The results of the experiment show that removing the RPN from Faster R-CNN has a positive impact on the mAP. This is caused by the model proposing many more object proposals to the detection head. However, the computational complexity increases significantly as well. Table 2 shows that removing the RPN increases the computational complexity by 17-fold. Considering that Faster R-CNN is already considered a slow and computationally complex model, removing the RPN in this manner is unfeasible and inoperable in real-world applications.

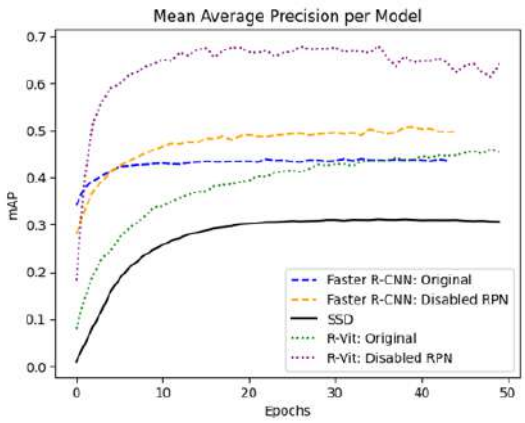


Figure 6: RPN Significance Test

The R-ViT model, which uses a Swin-FPN backbone shows similar results to Faster R-CNN(ResNet-50).

5.1.2 *SSD Comparison*

Table 2 shows that SSD is indeed computationally more efficient than Faster R-CNN as implied, despite nearly doubling the trainable parameters of Faster R-CNN. The results of this comparison show that SSD underperforms compared to Faster R-CNN in terms of mAP. Taking in mind the similarities between SSD and Faster R-CNN, this experiment suggests a positive impact of a region proposal stage in object detection models.

5.1.3 *Conclusion RPN Significance*

In summary, disabling the RPN in Faster R-CNN boosts mAP by increasing object proposals, but at a vast increase in computational complexity. Given Faster R-CNN's existing computational demands, this makes RPN removal impractical for real-world use. Comparatively, (SSD) outperforms Faster R-CNN in computational efficiency. However, SSD lags in mAP, highlighting the significance of a region proposal stage. This, in combination with prior research on the significance of the RPN, underscores the positive impact of implementing a distinct region proposal stage.

5.2 *Model construction*

5.2.1 *Hyperparameter Tuning Faster R-CNN*

Anchor Size
Results indicate a notable reduction in loss for the RPN when employing

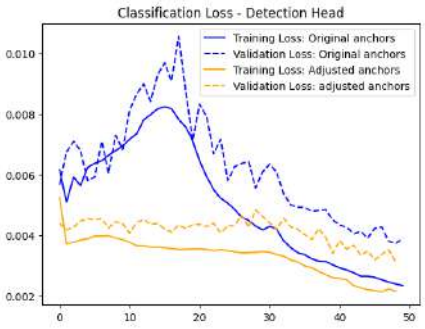
Model	FLOPs fwd	FLOPs fwd+bwd	MACs fwd	MACs fwd+bwd
Faster R-CNN (ResNet-50) RPN	268.28 GLOPS	804.83 GFLOPS	134.04 GMACs	402.13 GMACs
Faster R-CNN (ResNet-50) Disabled RPN	4.72 TFLOPS	14.17 TFLOPS	2.36 TMACs	7.09 TMACs
SSD (VGG16)	63.21 GFLOPS	189.62 GFLOPS	31.57 GMACs	94.72 GMACs
SSD (Swin-B)	82.64 GFLOPS	247.91 GFLOPS	41.26 GMACs	123.78 GMACs
Faster R-CNN (Swin-b FPN) RPN	619.67 GFLOPS	1.86 TFLOPS	309.43 GMACs	928.28 GMACs
Faster R-CNN (Swin-b FPN) Disabled RPN	69.18 TFLOPS	207.55 TFLOPS	34.57 TMACs	103.71 TMACs

Table 2: Computational Metrics for Various Models

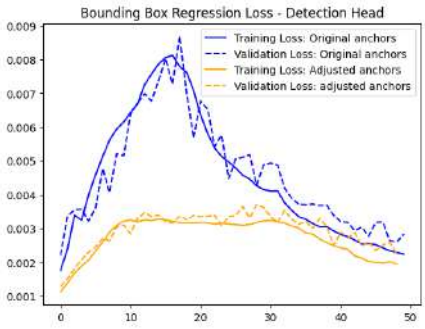
anchor box sizes specifically customized for a ViT backbone. In both scenarios, the RPN overfits the data after 30 epochs. However, this effect is less severe when utilizing adjusted anchor box sizes. The most significant differences occur in the RPN. This is potentially attributed to its enhanced ability to detect smaller objects when using smaller bounding boxes compared to the original anchor box sizes.
When considering the detection head losses in figure 7, using adjusted anchor box sizes stabilizes the training process and results in lower losses. Notable is that the loss of the original anchor box sizes only rises in the first 15 epochs. A plausible explanation is that the detection head needs to learn to account for poor object proposals from the RPN. This does not happen when using the adjusted anchor boxes.

RPN Convolutional Depth

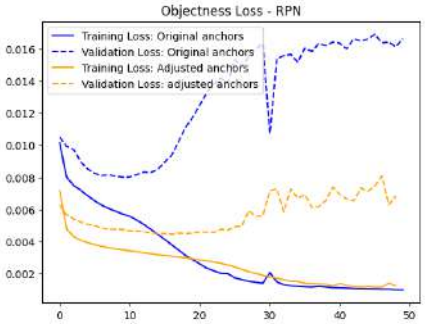
Results show that increasing the convolutional depth has the most impact on the RPNs' bounding box loss and objectness loss. Considering the objectness loss on the validation set, the results show that increasing the convolutional depth to 2 enhances the model slightly. The difference between a convolutional depth of 2 and 3 is negligible. Though the increase in convolutional depth has a positive effect on the validation loss, when considering the training loss as well, the model seems more prone to overfit on the training set when using a deeper RPN. The bounding box loss of the RPN shows the same effect. The losses of the detection head show no



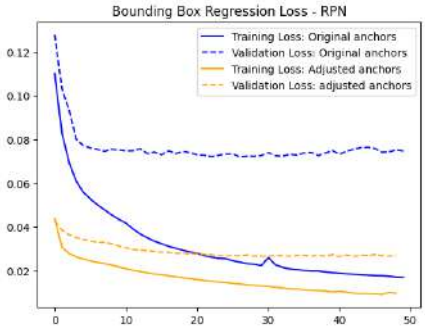
(a) Classification Loss - Detection Head



(b) Box Regression Loss - Detection Head

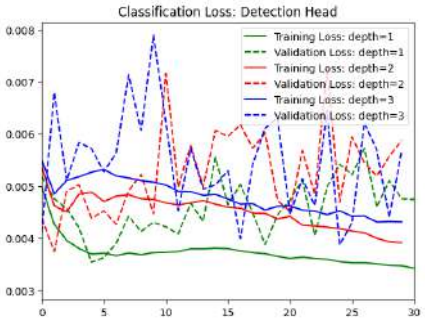


(c) Objectness Loss - RPN

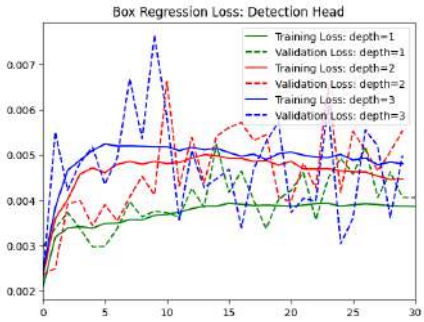


(d) Box Regression Loss - RPN

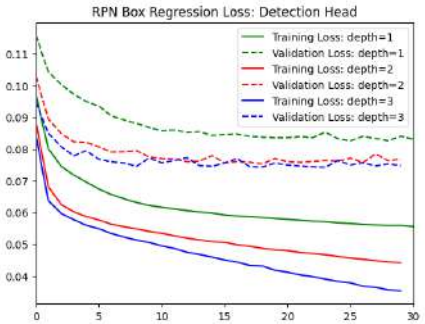
Figure 7: Experiment: Anchor Boxes



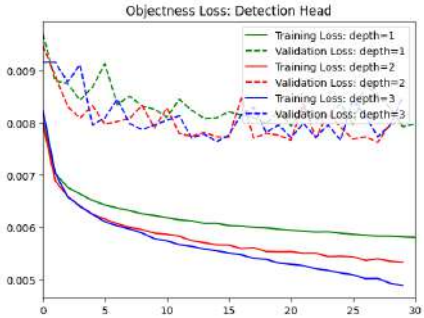
(a) Classification Loss - Detection Head



(b) Box Regression Loss - Detection Head



(c) Objectness Loss - RPN



(d) Box Regression Loss - RPN

Figure 8: Experiment: Convolutional Depth

clear sign of change when using varying convolutional depths.

Concluding Faster R-CNN Hyperparameters
The specified hyperparameters will be applied to the model in the upcoming experiments. The anchor sizes used in the following model will be proportional to the output size of the feature map. $[2^2, 4^2, 8^2, 12^2, 14^2]$ for feature maps of size 14^2, $[2^2, 4^2, 8^2, 16^2, 32^2]$ for feature maps of size 32^2. A convolutional depth of 2 will be used in the upcoming models. While increasing the convolutional depth positively affects the validation loss, the model is also more prone to overfitting. Techniques to prevent overfitting will be used to address this problem.

5.2.2 *Hyperparameter Tuning Backbone*

This section will reflect on experiments on optimizing the ViT backbone for object detection with a Faster R-CNN model.

Fine-tuning depth
Results show that the RPN is affected by adjusting the fine-tuning size of the backbone in how prone it is to overfitting on the training dataset. The box regression loss shows that the model is less prone to overfitting when fine-tuning fewer backbone layers. The objectness loss shows similar results. This suggests that the model overfits on the training data, but it will take more epochs for this effect to show.
The losses observed in the detection head indicate initial training instability during the early epochs. However, training and validation losses stabilize beyond a certain epoch threshold and exhibit a decreasing trend. This stabilization coincides with the epochs where the RPN begins to show overfitting. Considering that the detection head losses are affected and are directly influenced by the RPN's capability, the initially promising results from the detection head may be neglected. Consequently, it can be inferred that fine-tuning the backbone, particularly when utilizing a pre-trained ViT on ImageNet1k as the backbone for Faster R-CNN, is not a necessary step and does not yield a significant enhancement in the model's detection capabilities.

Backbone Size
The box regression loss for the RPN shows that using a ViT as a backbone is beneficial for localizing objects. Also, the smaller model is less prone to overfitting on the training data. On the contrary, the objectsenss loss for the RPN shows that the bigger models are better in predicting if there is an object at a particular anchor box.
The losses for the detection head suggest that the smaller model is superior

to the bigger models in classifying and localizing objects. When using the ViT-h/14 as a backbone, the model does not seem to be able to converge. Considering the mAP, the ViT-l/16 model emerges as the top performer, achieving a score of 0.35. The highest mAP score on the validation set for the ViT-b/16 model is 0.23. The primary distinction between the mAP scores of the two models is due to the ability to detect small objects, whereas the mAP score for small objects on the ViT-l/16 doubles that of the ViT-b/16. The mAP scores for medium and large objects show similar results in the ViT-b/16 and the ViT-l/16 model backbone.
Similar to the evaluation of the losses, the ViT-h/16 model could not converge and could not detect objects consistently.
Increasing the backbone size leads to increase model complexity and thus increased inference speed.

Model	FLOPs fwd	FLOPs fwd+bwd	MACs fwd	MACs fwd+bwd
Faster R-CNN (ViT-b/16)	94.76 GLOPS	282.27 GFLOPS	47.36 GMACs	142.09 GMACs
Faster R-CNN (ViT-l/16)	577.95 GFLOPS	1.73 TFLOPS	288.79 GMACs	877.37 GMACs
Faster R-CNN (ViT-h/14)	1.27 TLOPS	3.82 TFLOPS	636.34 GMACs	1.91 TMACs

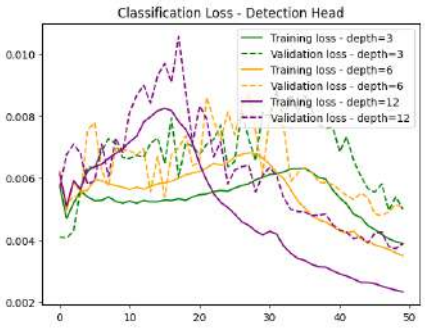
Table 3: Computational Metrics for Various Models

Concluding
The training losses suggest that using the smallest ViT implementation is the best fit for the Faster R-CNN architecture, however, the mAP shows that increasing the backbone to a ViT-l/16 can be beneficial. This is most likely due to the ViT-l/16 having a bigger input size and is, therefore, more able to capture finer details in the images. Notably, increasing the backbone size greatly increases the model's complexity and thus inference speed for the model.

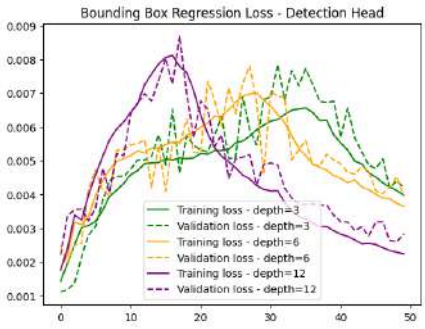
5.2.3 *Hyperparameter Tuning Swin*

Plain Swin Transformer Backbone: Faster R-CNN (Swin)

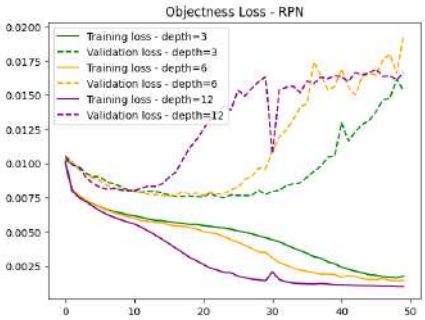
The validation and training losses for the RPN of Faster R-CNN(Swin) show that the model is not able to generalize on the training data. The RPN objectness loss and box regression loss are considerably higher than their ViT counterparts. This suggests that the model is significantly worse at localizing objects. Also, the losses for the classification head are notably higher than the ViT implementation. Furthermore, the model does not converge as the losses keep increasing keep increasing over time.
The mAP on the validation set suggests that the model does not generalize



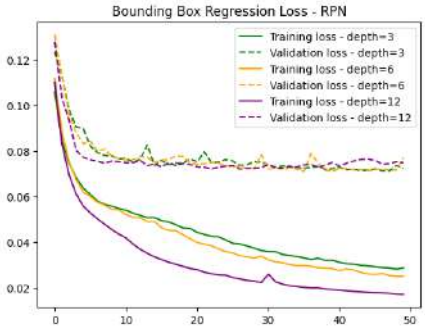
(a) Classification Loss - Detection Head



(b) Box Regression Loss - Detection Head

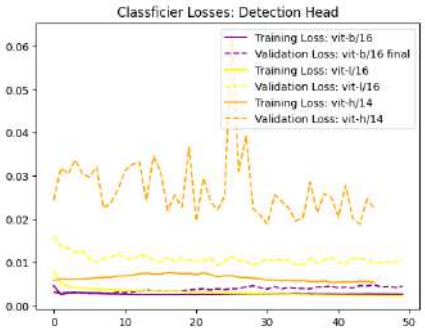


(c) Objectness Loss - RPN



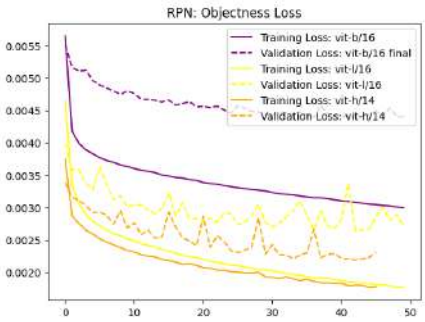
(d) Box Regression Loss - RPN

Figure 9: Experiment: Finetuning Backbone Depth

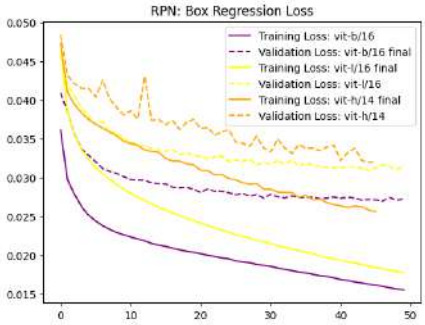


(a) Classification Loss - Detection Head

(b) Box Regression Loss - Detection Head



(c) Objectness Loss - RPN



(d) Box Regression Loss - RPN

Figure 10: Experiment: Backbone Size

well on the validation set and is not able to make any predictions.
Anticipatedly, this outcome is expected, given that the feature map generated by the Swin Transformer is even more compact than the output of the ViTs. Despite the Swin Transformer's capability to capture finer details from an input image, it appears that the Faster R-CNN object detection model is not able to exploit these details. Thus, making it unfit to serve as a backbone for Faster R-CNN.

Swin FPN Backbone: Faster R-CNN(Swin-FPN)
Adding an FPN significantly improves the model's ability to detect objects, considering the validation and training losses. In addition to the initial lower losses observed for both the RPN and the detection head, the model successfully converges. Also, comparing the mAP of Faster R-CNN (Swin) and Faster R-CNN (Swin-FPN) show a significant increase.

Swin Size
In the comparison of the Swin Transformer sizes, the implementation of Swin-t and Swin-s yields comparable outcomes, whereas the Swin-b implementation deviates. The Swin-b implementation shows lower initial loss, and faster divergence in all losses but the RPN box regression loss.
When examining the mAP, the results mirror the observed trends in the losses. The Swin-b model outperforms the Swin-s and Swin-t model. Notably, this distinction is particularly evident in the detection of small and medium-sized objects. The Swin-s and Swin-t model perform similarly. The detection of large objects remains consistent across all three models. This suggests that increasing the backbone size is beneficial for the model ability to detect smaller objects.
Table 4 shows the models computational complexity. The difference in complexity is greatest when increasing the model from Swin-t to Swin-s. Consequently, the adoption of Swin-S may not be particularly advantageous, as it results in a twofold increase in model complexity with a relatively modest impact on mAP compared to Swin-t. However, when progressing to Swin-b, a considerable effect becomes apparent.

Concluding Swin Transformer Backbone
Faster R-CNN(Swin) struggles to generalize on both training and validation data, exhibiting higher losses in objectness, box regression, and classification compared to its ViT counterparts. The model's inability to converge and low mAP on the validation set further indicate its limitations. The addition of an FPN significantly enhances object detection capabilities, leading to lower losses and successful convergence. When comparing different Swin Transformer sizes, Swin-b outperforms Swin-s and Swin-t,

Model	FLOPs fwd)	FLOPs fwd+bwd)	MACs fwd	MACs fwd+bwd
Faster R-CNN (Swin-t/FPN)	316.98 GLOPS	950.93 GFLOPS	158.26 GMACs	474.79 GMACs
Faster R-CNN (Swin-s/FPN)	692.36 GFLOPS	2.08 TFLOPS	345.78 GMACs	1.04 TMACs
Faster R-CNN (Swin-b/FPN)	878.98 GLOPS	2.63 TFLOPS	438.01 GMACs	1.31 TMACs

Table 4: Computational Metrics for Various Swin-FPN Backbones for Faster R-CNN

especially in detecting small and medium-sized objects, highlighting the advantage of increasing backbone size. Additionally, employing Focal Loss in the model results in slightly lower losses across metrics and a modest improvement in mAP compared to using the original loss.

5.2.4 *Additional Optimizations*

Focal Loss
Upon comparing the losses between the model employing Focal Loss and the one utilizing the original loss, it is evident that the former exhibits marginally lower loss values across all four metrics. Considering the mAP, replacing the original loss with Focal loss poses a slight improvement.

Transformer RPN
The model employing an RPN incorporating a transformer encoder appears to underperform when evaluating its losses. Across all loss measures, the transformer RPN model exhibits higher losses. However, when assessing the mAP score for both models, the transformer RPN model surpasses the model original RPN model by approximately 0.02. This indicates that replacing the convolutional layers of the RPN for transformer layer can have a positive influence on the precision of the models.

Transformer Detection Head
Upon substituting the convolutional layer in the detection head, the outcomes indicate a greater loss across all measures. Furthermore, the training loss for the detection head decreases gradually, while the validation loss plateaus after approximately 25 epochs. This implies that the model is overfitting the training data after 25 epochs. In contrast, this is not observed in implementations that exclude transformer models from their detection head. This discrepancy implies that the replacement of convolutional layers in the detection head may lead the model to be more prone to overfitting. The mAP shows that the model utilizing a transformer detection head demonstrates a faster divergence towards a minimum. Nonetheless, given

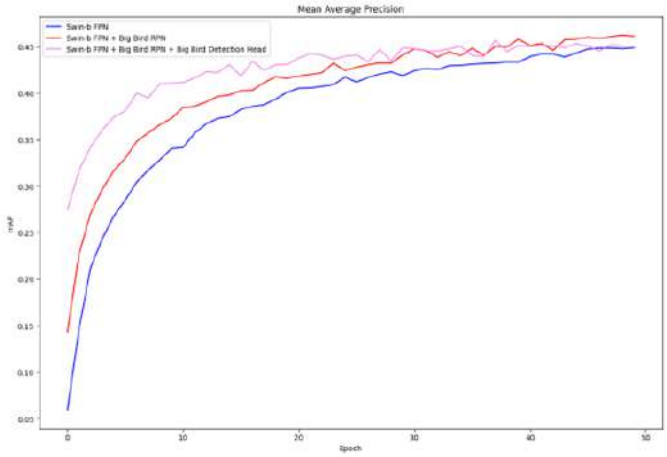


Figure 11: Experiment Additional Transformers

enough training epochs, it achieves comparable performance to the model without transformers (excluding the backbone). It underperforms compared to the model that uses transformers only in its RPN.
Considering that replacing the convolutional layers for transformer encoder layers significantly adds complexity to the model, that the models' losses show that adding transformers in the detection head causes the model to be more prone to overfit, and that the mAP does not increase significantly compared to less complex models, it can be concluded that replacing the convolutional layers in the detection head for transformer encoder layers is not beneficial for the models performance.

5.3 *Models comparison based on Metrics*

Table 5 shows that the Faster R-CNN (Swin-FPN) is the best-performing model. Considering the overall mAP, this model outperforms the Faster R-CNN (ResNet-50-FPN) model by 0.03. This is mainly caused by the models' ability to detect small objects. Faster R-CNN (ResNet-50-FPN) outperforms its Swin counterpart on medium and large objects. The mAR values show similar results.
Considering the relative decrease in mAP (IoU > 0.5) and the mAP(IoU > 0.75), the Swin-FPN backbone shows a smaller decrease in mAP (-25%) than the ResNet-50-FPN backbone (-28%). Also, the mAP values for both the IoU thresholds are considerably higher for the Swin-FPN backbone. This suggests that the Swin-FPN model is better at localizing objects.
The ViT backbone shows the lowest results on both mAP and mAR. The most significant difference is found in the small objects, where the mAP

score for small objects is 0.05, and the mAR for small objects is 0.19. These values show that this model is unsuitable for object detection on small objects. While considering the mAP and mAR for medium and large objects, the ViT backbone outperforms the Swin and the Resnet-50 implementation. However, it can be noted that the values for the medium and large objects are considerably s and show that the ViT backbone is able to detect objects.
A comparison between Faster R-CNN(Swin-FPN) and YOLOv8 shows that Faster R-CNN(Swin-FPN) outperformrs YOLOv8. The most significant difference is in the small objects where Faster R-CNN(Swin-FPN) exceeds by ~22 percent points. The other metrics show similar results between the two models. This indicates that YOLOv8 is not fitted for small object detection, and Faster R-CNN(Swin-FPN) outperforms it on mAP.

Model	**ResNet-50-FPN**	**Swin-b-FPN**	**ViT-h/16**	**YOLOv8**
mAP	0.45	0.48	0.33	0.37
mAP (IoU > 50)	0.68	0.73	0.60	0.50
mAP (IoU > 75)	0.49	0.54	0.34	0.41
mAP Small	0.24	0.29	0.05	0.07
mAP Medium	0.54	0.42	0.26	0.26
mAP Large	0.62	0.55	0.47	0.50
mAR 1	0.34	0.34	0.27	0.30
mAR 10	0.55	0.59	0.42	0.44
mAR 100	0.59	0.63	0.45	0.44
mAR Small	0.43	0.49	0.19	0.10
mAR Medium	0.67	0.59	0.41	0.34
mAR Large	0.72	0.65	0.58	0.57

Table 5: mAP & mAR for Faster R-CNN backbones and for YOLOv8

5.4 *Error analysis*

5.4.1 *Over-under predictions*

Figure 12 shows the underprediction and overprediction proportional to the number of predictions per class. Overprediction occurs when a model detects an object, but there is no actual object. For each cell, the proportional overprediction γ_{over_c} is defined as $\gamma_{over_c} = \frac{O_c}{P_c}$ where O is the number of overclassifications, P is the total number of predictions, and c represents the class. Underprediction occurs when there exists an object, but the model fails to detect it. For each cell, the proportional underprediction

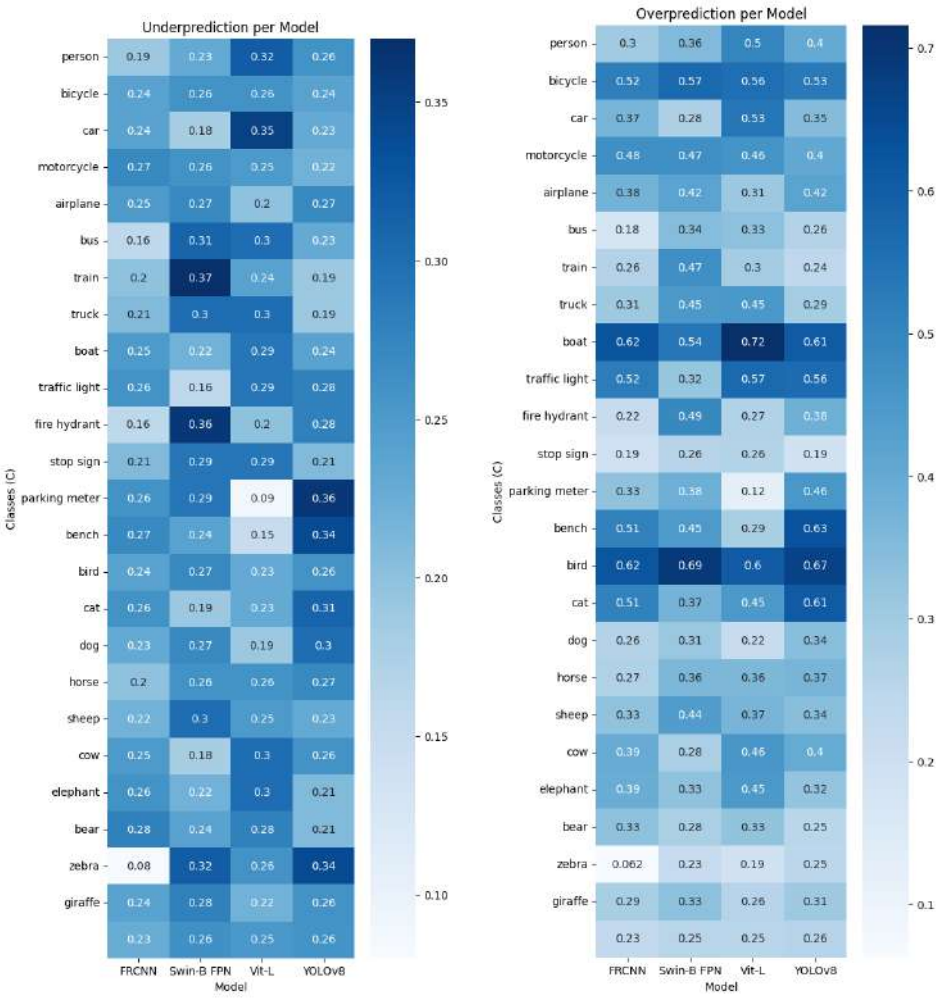
γ_{under_c} is defined as $\gamma_{under_c} = \frac{U_c}{P_c}$ where U is the number of under classifications, P is the number of predictions, and c is the class. A model performs optimally when it minimizes both under and overpredictions.
The figure shows no clear pattern when considering the underpredictions per class. Though, when looking at the overpredictions per class, the classes bird, boat, and bicycle have a higher percentage of overpredictions, while bear giraffe and zebra have a lower percentage underpredictions. There is no clear pattern when considering difference in models.
Table 6 shows the percentage of underprediction, overpredictions, misclassifications and correct classifications for all models. This shows that Faster R-CNN(ResNet-50-FPN) has the lowest percentage of over and underprediction, close is Faster R-CNN(Swin-FPN). Faster R-CNN(ViT-l/16) shows to perform the worst.

Model	**Faster R-CNN**	**Swin**	**ViT**	**YOLO**
Under Predicted	0.254	0.271	0.310	0.282
Over Predicted	0.264	0.274	0.349	0.303
Misclassified	0.014	0.016	0.004	0.036
Correct (acc)	0.468	0.440	0.337	0.380

Table 6: Prediction Analysis for Various Models

5.4.2 *Misclassification per class*

Misclassifications occur when the model identifies an object where an actual object exists but is unable to classify the object correctly. Figure 13 shows misclassification matrices for all models, highlighting only the misclassification by excluding underpredictions, overpredictions and correct predictions (diagonal). Lines are added to subdivide classes into their supercategories. Also, the matrix is row normalized to improve readability. All matrices show similar patterns of misclassification, with the most misclassifications occuring in classes that share supercategories. This consistent pattern suggests that the models, struggle with differentiating objects within the same supercategory. This outcome is expected, as classes that share supercategories tend to share more similarities than objects that do not share supercategories.
Notably, all models, except for ViT, exhibit the highest misclassification for the predicted class "person," followed by "car" as the second most frequent error. Even after class balancing, “person” and “car” remain the majority classes, suggesting that the models may be overfitting on these prevalent categories. The findings indicate that YOLOv8 is particularly susceptible

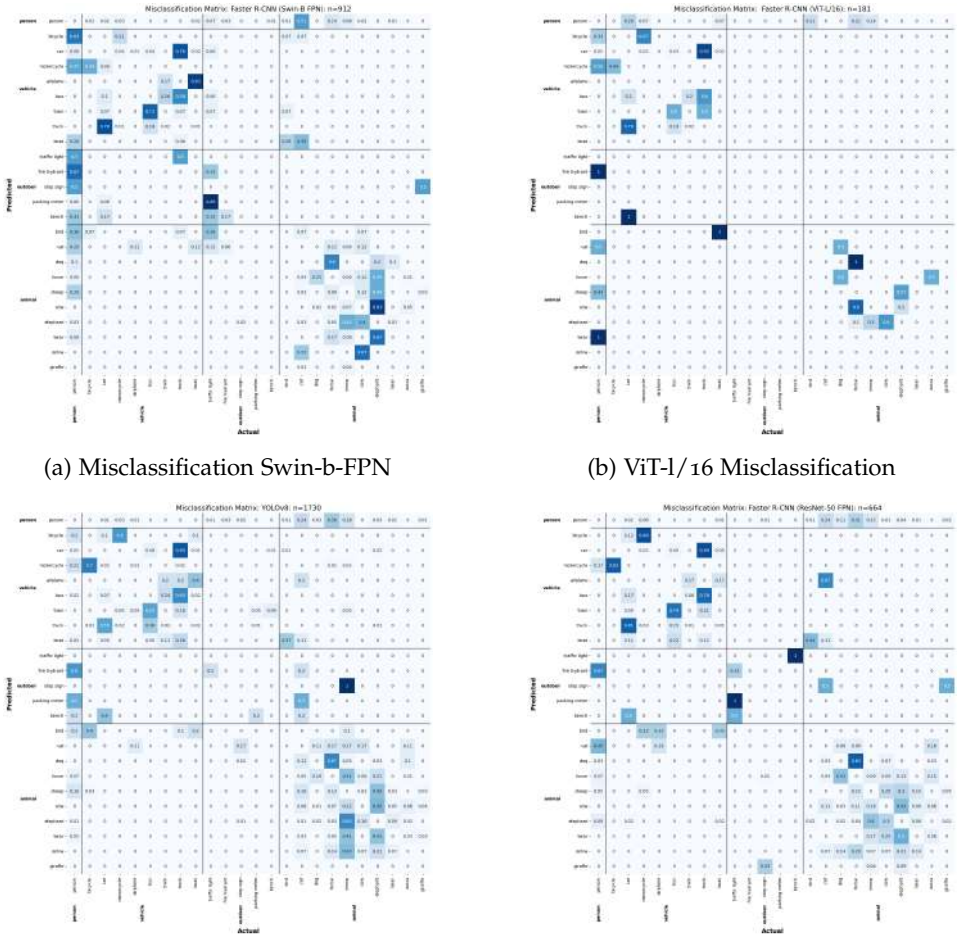


(a) Underprediction (b) Overprediction

to overfitting on majority classes compared to the Faster R-CNN-based models.

5.4.3 *Number of Objects*

Figure 14 illustrates that images with fewer objects demonstrate a higher mAP, a trend observed across all models. Faster R-CNN (ViT) and YOLOv8 show similar results. Their mAP at one object per image is the lowest of all models. Also, their mAP has the steepest descent with an increasing number of objects. Although Faster R-CNN (ResNet-50-FPN) attains the highest mAP score with one object per image, it experiences a sharp decline. After eight objects per image, the mAP for the Faster R-CNN (Swin-FPN) model is the highest.
These findings suggest that for scenarios involving a limited number of



(a) Misclassification Swin-b-FPN

(b) ViT-l/16 Misclassification

(c) YOLOv8 Misclassification

(d) Faster R-CNN Misclassification

Figure 13: Misclassification Matrices

objects in a single image, Faster R-CNN (ResNet-50-FPN) is the most suitable choice. Conversely, Faster R-CNN (Swin-FPN) would perform better in scenarios with numerous objects in a single image.

6 DISCUSSION

6.1 *Goal and results*

The goal of this research is to determine the significance of the RPN in object detection models and to enhance two-stage object detection models by incorporating transformers, aiming for a high-precision object detection model.
The results showed that incorporating transformers comes with its chal-

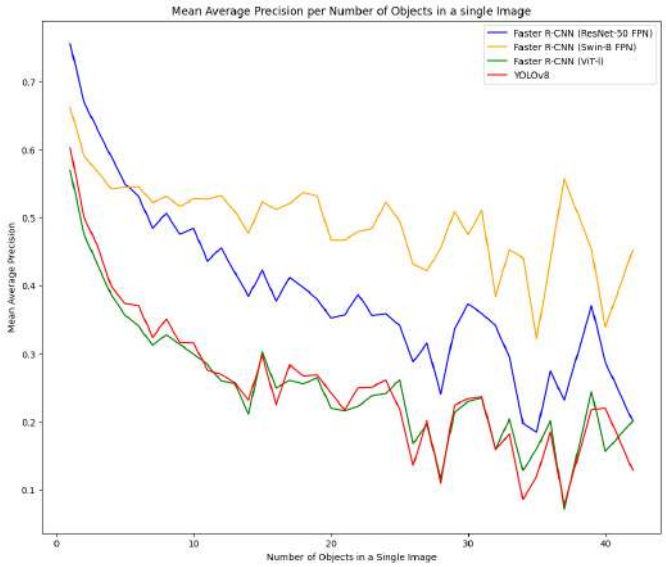


Figure 14: Number of Objects per Image

lenges. Faster R-CNN can be optimized using transformer-based models as a backbone. When correctly applying these optimizations, transformer-based backbones prove to be able to improve on the original Faster R-CNN implementation, which incorporates CNN-based backbones. Also, the transformer-enhanced implementation of Faster R-CNN outperforms the state-of-the-art model YOLOv8 on precision and recall. Notably, the mAP for YOLOv8 achieved in this study is much lower than its performance listed on the COCO-test-dev page.

6.2 *Literature reflection*

Raghu et al. state that ViTs are better at localization than CNNs, primarily attributed to the addition of positional embeddings and a distinct classification token. The comparison Faster R-CNN (ResNet-50) and the transformer-based backbones reveals that when increasing the IoU, the mAP for ResNet-50 shows a more pronounced decline. This indicates that the transformer-based model outperforms CNNs on localization.
Small object detection can be challenging when doing object detection. Local features are necessary to detect small objects. Raghu et al. shows that when using ViTs pre-trained only on ImageNet-1k, ViTs are not able to attend to local features. A newer method introduces Shifted Windows that structures ViTs into a hierarchical network. (Z. Liu, Hu, et al., 2021; Z. Liu, Lin, et al., 2021). This research showed that incorporating the Swin Transformer's hierarchical structure allows for better object detection.

As demonstrated in (Y. Li et al., 2022), incorporating a pyramid structure within the backbone substantially enhances the model's object detection capabilities. This is evident in the comparison between Swin with and without a FPN. (T.-Y. Lin et al., 2016) While the authors of the mentioned paper indicate that adding a FPN results in only a marginal improvement in detection capabilities, this study underscores the necessity of incorporating a pyramid structure when utilizing a Swin Transformer as the backbone for Faster R-CNN. This necessity arises from the smaller outputted feature map and the inherent anchor generation characteristics of the RPNs.
ViTs have proven beneficial for many computer Vision Tasks. (Cuenat & Couturier, 2022; Filipiuk & Singh, 2022; Maurício et al., 2023),
This indicates that ViTs could be used as a backbone for object detection tasks. However, specific challenges emerge in evaluating ViTs against CNNs as the foundation for object detection models. However, upon refining the detection head with the transformer backbone, it becomes evident that object detection can also gain advantages from ViTs. The out-of-sample comparison between Faster R-CNN(ResNet-50-FPN) and Faster R-CNN(Swin-FPN) shows that Faster R-CNN can perform better when utilizing a transformer-based backbone.

6.3 *Impact*

This study provides evidence supporting the use of transformer-based feature extractors as a backbone for object detection models, avoiding the need for significantly larger image sizes or patch-padding methods that would demand a considerable increase in computational power. However, incorporating plain ViTs as a backbone for Faster R-CNN is not an optimal solution. This study shows that ViTs can successfully work as a backbone for object detection, thus paving the way for further research into transformer-based backbones for object detection. Moreover, this research shows that a Swin-FPN structure can work exceptionally well as an object detection backbone due to its hierarchical structure and ability to capture local and global features. Thus making it versatile across multiple object sizes. Considering the overall structure of Faster R-CNN with transformer-based models, the study outlines the necessary adjustments to make Faster R-CNN compatible with transformer-based models.

6.4 *Limitations and future directions*

Given that one of the primary objectives of this study was to aim for a high-precision object detection model, a comparative analysis with the current state-of-the-art models is crucial. In this study, the YOLOv8 model

is used as a comparison. On the full COCO-test-dev dataset, the model is able to achieve a mAP of over 0.54. However, when trained on a subset of the COCO dataset, notably lower results are obtained. More comparisons should be made to enhance comparisons with the state-of-the-art models. An example would be CO-DETR, currently the highest-scoring object detection model on the COCO-test-dev dataset.
An alternative approach to assess the model's generalization capabilities involves evaluating it on the COCO-test-dev dataset. Therefore, the model must be trained on the full COCO dataset and evaluated via the COCO-test-dev server. Unfortunately, this was not feasible in the scope of this research due to constraints in time and computational resources.

7 CONCLUSION

7.1 *What is the impact of the RPN in object detection models?*

Prior research has proven the significance of the RPN. When assessing the effect of the RPN in Faster R-CNN, it shows benefits in precision and speed. Comparing the model to a single-stage object detection model, it outperforms this model in precision. Conversely, when comparing the Faster R-CNN model with a disabled RPN, the model with a disabled RPN outperforms the original model in terms of precision, suggesting a potential disadvantage of the RPN. However, disabling the RPN leads to an excessive number of proposals, rendering the model impractical for real-time scenarios. In conclusion, the RPN in Faster R-CNN proves beneficial in constraining region proposals and enhancing precision, but it also contributes positively to mAP when compared to single-stage object detection models.

7.2 *What is the effect of enhancing Faster R-CNN using transformers?*

When evaluating the precision and recall, Faster R-CNN (Swin-FPN) demonstrates superiority over Faster R-CNN (ResNet-50-FPN). The performance is notably enhanced, particularly in scenarios involving smaller objects. While ViTs can function adequately as a backbone for Faster R-CNN, it is important to note that they may not be as effective within the framework of Faster R-CNN. The implementation of transformers in the RPN and the detection head was shown to have minor effects on the models' performance.
When compared to the state-of-the-art model YOLOv8, the utilization of a transformer-based backbone, specifically the Swin-FPN variant, showcases improved performance in Faster R-CNN, surpassing the capabilities

of YOLOv8. This advantage is particularly evident in detecting smaller objects, where YOLOv8 cannot detect objects consistently. It is essential to note that YOLOv8 does not perform as effectively as proven in other studies, rendering this result unreliable for conclusions.

7.3 *What are the most prevalent types of errors in object detection with R-VIT?*

The most prevalent types of errors observed in object detection are consistent across all models, indicating similar outcomes regarding over/under prediction and misclassification. Notably, the most common misclassifications predominantly occur within the same supercategory. Also, when considering the majority classes in the misclassifications, all models tend to overfit the majority classes. YOLOv8 shows the most severe overfitting. It's worth highlighting that Faster R-CNN (Swin-FPN) shows the least errors when dealing with scenarios involving multiple objects, demonstrating superior performance in such cases.

7.4 *Overarching RQ: How does the performance of R-ViT, which incorporates transformer models in Faster R-CNN, compare to state-of-the-art object detection?".*

In summary, the RPN in Faster R-CNN has been proven to be beneficial for object detection. Faster R-CNN incorporating transformer-based backbones demonstrates superior performance when employing appropriate hyperparameters, particularly excelling in images with numerous objects. It surpasses the original Faster R-CNN with ResNet50-FPN. The most promising results are achieved when utilizing the Swin-FPN backbone. Incorporating transformers further in the network has a minor effect on the overall precision.
When compared with the state-of-the-art object detection model YOLOv8, R-ViT demonstrates promising outcomes, particularly in the detection of small objects. Despite achieving noteworthy success, it lags behind other existing state-of-the-art models in terms of both speed and mAP. As a result, Faster R-CNN object detection models that implement transformer models are unlikely to rival the existing state-of-the-art object detection models.

REFERENCES

Beal, J., Kim, E., Tzeng, E., Huk, D., Andrew, P., Dmitry, Z., & Pinterest, K. (2020). Toward transformer-based object detection. https://arxiv.org/abs/2012.09958v1

Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., & Zagoruyko, S. (2020). End-to-end object detection with transformers. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *12346 LNCS*, 213–229. https://doi.org/10.1007/978-3-030-58452-8_13

Cuenat, S., & Couturier, R. (2022). Convolutional neural network (cnn) vs vision transformer (vit) for digital holography. *2022 2nd International Conference on Computer, Control and Robotics, ICCCR 2022*, 235–240. https://doi.org/10.1109/ICCCR54399.2022.9790134

Deininger, L., Stimpel, B., Yuce, A., Abbasi-Sureshjani, S., Schönenberger, S., Ocampo, P., Korski, K., & Gaire, F. (2022). A comparative study between vision transformers and cnns in digital pathology. https://arxiv.org/abs/2206.00389v1

Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *NAACL HLT 2019 - 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference*, *1*, 4171–4186. https://arxiv.org/abs/1810.04805v2

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., & Houlsby, N. (2020). An image is worth 16x16 words: Transformers for image recognition at scale. *ICLR 2021 - 9th International Conference on Learning Representations*. https://arxiv.org/abs/2010.11929v2

Filipiuk, M., & Singh, V. (2022). Comparing vision transformers and convolutional nets for safety critical systems. https://www.tensorflow.org/datasets/catalog/imagenet2012

Huang, J., Rathod, V., Sun, C., Zhu, M., Korattikara, A., Fathi, A., Fischer, I., Wojna, Z., Song, Y., Guadarrama, S., & Murphy, K. (2016). Speed/accuracy trade-offs for modern convolutional object detectors. *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, *2017-January*, 3296–3305. https://doi.org/10.1109/CVPR.2017.351

Kim, J. A., Sung, J. Y., & Park, S. H. (2020). Comparison of faster-rcnn, yolo, and ssd for real-time vehicle type recognition. *2020 IEEE*

International Conference on Consumer Electronics - Asia, ICCE-Asia 2020. https://doi.org/10.1109/ICCE-ASIA49877.2020.9277040

Li, M., Zhang, Z., Lei, L., Wang, X., & Guo, X. (2020). Agricultural greenhouses detection in high-resolution satellite images based on convolutional neural networks: Comparison of faster r-cnn, yolo v3 and ssd. *Sensors 2020, Vol. 20, Page 4938*, *20*, 4938. https://doi.org/10.3390/S20174938

Li, Y., Mao, H., Girshick, R., & He, K. (2022). Exploring plain vision transformer backbones for object detection. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *13669 LNCS*, 280–296. https://doi.org/10.1007/978-3-031-20077-9_17

Lin, T. Y., Goyal, P., Girshick, R., He, K., & Dollar, P. (2017). Focal loss for dense object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *42*, 318–327. https://doi.org/10.1109/TPAMI.2018.2858826

Lin, T.-Y., Dollár, P., Girshick, R., He, K., Hariharan, B., & Belongie, S. (2016). Feature pyramid networks for object detection. https://arxiv.org/abs/1612.03144v2

Lin, T., Maire, M., Belongie, S. J., Bourdev, L. D., Girshick, R. B., Hays, J., Perona, P., Ramanan, D., Doll'a r, P., & Zitnick, C. L. (2014). Microsoft COCO: common objects in context. *CoRR*, *abs/1405.0312*. http://arxiv.org/abs/1405.0312

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., & Berg, A. C. (2015). Ssd: Single shot multibox detector. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *9905 LNCS*, 21–37. https://doi.org/10.1007/978-3-319-46448-0_2

Liu, Z., Hu, H., Lin, Y., Yao, Z., Xie, Z., Wei, Y., Ning, J., Cao, Y., Zhang, Z., Dong, L., Wei, F., & Guo, B. (2021). Swin transformer v2: Scaling up capacity and resolution. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, *2022-June*, 11999–12009. https://doi.org/10.1109/CVPR52688.2022.01170

Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., & Guo, B. (2021). Swin transformer: Hierarchical vision transformer using shifted windows. *Proceedings of the IEEE International Conference on Computer Vision*, 9992–10002. https://doi.org/10.1109/ICCV48922.2021.00986

Loshchilov, I., & Hutter, F. (2017). Decoupled weight decay regularization. *7th International Conference on Learning Representations, ICLR 2019*. https://arxiv.org/abs/1711.05101v3

Maurício, J., Domingues, I., & Bernardino, J. (2023). Comparing vision transformers and convolutional neural networks for image classifi-

cation: A literature review. *Applied Sciences 2023, Vol. 13, Page 5521*, *13*, 5521. https://doi.org/10.3390/APP13095521

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Köpf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., … Chintala, S. (2019). Pytorch: An imperative style, high-performance deep learning library. *Advances in Neural Information Processing Systems*, *32*. https://arxiv.org/abs/1912.01703v1

Raghu, M., Unterthiner, T., Kornblith, S., Zhang, C., & Dosovitskiy, A. (2021). Do vision transformers see like convolutional neural networks? *Advances in Neural Information Processing Systems*, *15*, 12116–12128. https://arxiv.org/abs/2108.08810v2

Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2015). You only look once: Unified, real-time object detection. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, *2016-December*, 779–788. https://doi.org/10.1109/CVPR.2016.91

Redmon, J., & Farhadi, A. (2018). Yolov3: An incremental improvement. https://arxiv.org/abs/1804.02767v1

Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *39*, 1137–1149. https://doi.org/10.1109/TPAMI.2016.2577031

Tan, L., Huangfu, T., & Wu, L. (2021). Comparison of yolo v3, faster r-cnn, and ssd for real-time pill identiication. https://doi.org/10.21203/rs.3.rs-668895/v1

Terven, J. R., & Cordova-Esparza, D. M. (2023). A comprehensive review of yolo: From yolov1 and beyond. https://arxiv.org/abs/2304.00501v5

Vasconcelos, C., Birodkar, V., & Dumoulin, V. (2022). Proper reuse of image classification features improves object detection. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, *2022-June*, 13618–13627. https://doi.org/10.1109/CVPR52688.2022.01326

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. *Advances in Neural Information Processing Systems*, *2017-December*, 5999–6009. https://arxiv.org/abs/1706.03762v7

Zaheer, M., Guruganesh, G., Dubey, A., Ainslie, J., Alberti, C., Ontanon, S., Pham, P., Ravula, A., Wang, Q., Yang, L., & Ahmed, A. (2020). Big bird: Transformers for longer sequences. *Advances in Neural Information Processing Systems*, *2020-December*. https://arxiv.org/abs/2007.14062v2

Zong, Z., Song, G., & Liu, Y. (2022). Detrs with collaborative hybrid assignments training. https://arxiv.org/abs/2211.12860v6