Implementing and Training DETR for Custom Face Detection on WIDER Face Dataset

M Shilpa

Computer Science Department, IIT Tirupati

cs21b034@iittp.ac.in

Abstract

The project is based on implementing and training DETR(model)for custom dataset using wider face dataset. Here the DETR model is meant for finding the face in every image and detecting the face, the model uses transformers for detection which has encoder and decoder which helps in accurate detection. The model is trained and fine-tuned on custom dataset to optimize its performance for face dataset, the backbone for the model is ResNet-50, ResNet-50 serves as a robust backbone for the DETR model, enhancing its ability to detect and localize faces accurately within images(where it extracts the important features from the input image). Various experiments are conducted by changing the hyperparameters, to evaluate the impact of different hyperparameters. The results gives the potential of DETR in face detection tasks and provide insights into its application in real-world scenarios. This project will help in growing body of research on transformer-based models for object detection.

Acknowledgment

I would like to thank my guide Dr. chalavadi vishnu for guiding and supporting me throughout this project. Also iam greatful to Facebook AI for developing the DETR (Detection Transformer) model, which served as the foundation for my work on custom face detection. Iam very thankful to the authors of the WIDER Face dataset for providing the dataset which i used in this project,and i would like to thank cs22s503@iittp.ac.in, k.jahnavi who have helped me in doing this project.

About the Institute

IIT Tirupati is an autonomous Institute under the Ministry of Education, Government of India. IIT Tirupati is one of the Institutes of National Importance recognized by an Act in Parliament of India known as Institutes of Technology Act, 1961.

Academic Excellence and Research

IIT Tirupati has a commitment to create a strong academic setting that emphasizes research across disciplines and new ideas. The institute provides many programs for undergrads, postgrads, and doctoral students in different fields such as engineering, science, and humanities. The courses aim to give students a solid base in theory while pushing them to apply what they learn and think.

Student Development

IIT Tirupati makes student development a top priority. The institute offers many chances for students to take part in activities outside class, programs to build leadership skills, and internships. These experiences add to their academic journey and get them ready for successful careers in their fields. The lively campus life and supportive environment at the institute help students grow in all areas of their lives.

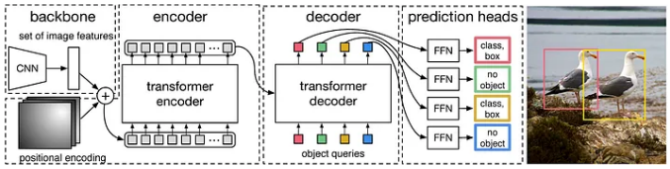
Future Vision

As we look to the future, IIT Tirupati is set to keep moving forward on its path to become a top global center for research and learning. The institute has big plans in mind. These include growing its research abilities teaming up with partners around the world, and making its classes even better. IIT Tirupati wants to leave its mark on the world. It aims to do this through its work in science, tech, and teaching. The institue is committed to being the best it can be and coming up with new ideas.

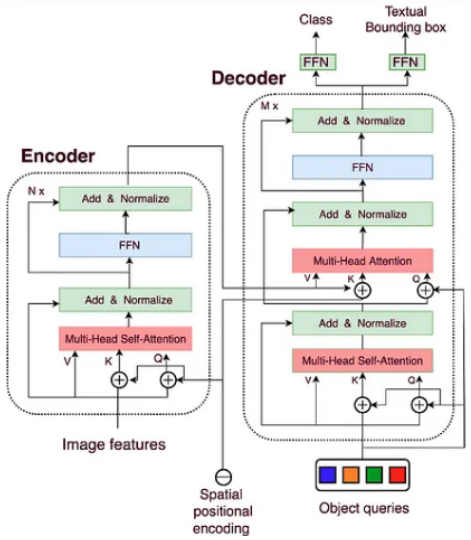
List of Figures

Figure 1: Architecture of DETR Model
The DEtection TRansformer (DETR) model is designed for end-to-end object detection. The model leverages a transformer-based approach to predict bounding boxes and class labels directly from image data, utilizing attention mechanisms for enhanced performance. The key components of the DETR architecture are as follows:

- **CNN Backbone:** Backbone extracts the important features from the input images, which are further sent to the transformer encoder.
- **Transformer Encoder:** The features are sent to the transformer encoder, and it processes these features to capture the relationships between different parts of the image using self-attention mechanisms.
- **Transformer Decoder:** The decoder makes use of those encoded capabilities to generate predictions (it includes the object to be detected and the bounding box for the object) through focusing on relevant parts of the input image. It also uses self-attention to focus on different parts of the image while making these predictions.



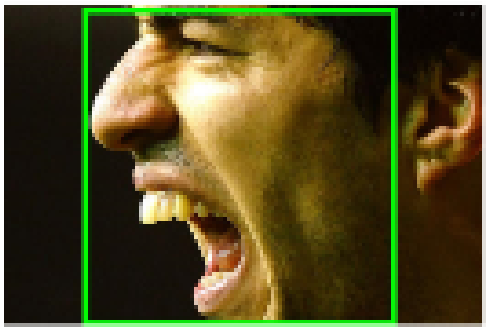
(a) DETR architecture



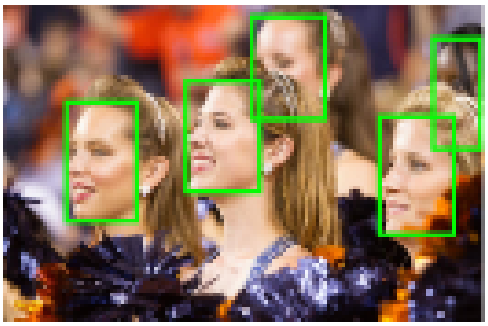
(b) Encoder-Decoder of Transformer

Figure 1: Architecture of DETR Model

- **Object Queries:** Object queries are learnable embeddings in model DETR that help the network identify and locate objects in an image. They guide the model to predict the presence, position, and category of each object.



(a) Image 1

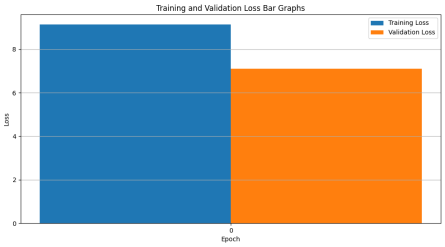


(b) Image 2

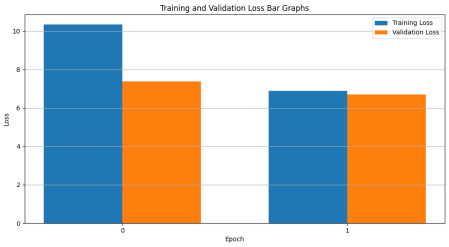
Figure 2: Images from the WIDER Face Dataset

Figure 2: WIDER Face Dataset
The WIDER FACE dataset is a face detection benchmark dataset, which is derived from the publicly available WIDER dataset. We selected 32,203 images and labeled 393,703 faces, capturing a high degree of variability in scale, pose, and occlusion, as illustrated in the sample images. The WIDER FACE dataset is organized into 61 event classes. For each event class, we randomly partition the data into training, validation, and testing sets, with 40%, 10%, and 50% of the data allocated to each set, respectively.

Figure 3: Training and Validation Loss
This figure illustrates the training and validation loss for two separate experiments.



(a) Experiment-1



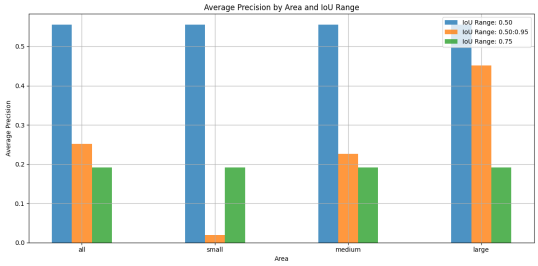
(b) Experiment-2

Figure 3: Training and Validation Loss

- **Experiment-1:** Here x-axis denotes the epoch and y-axis denotes the loss we can observe for the experiment-1, the training loss is 9.5(approx) and validation loss is 7(approx).
- **Experiment-2:** Here x-axis denotes the epoch and y-axis denotes the loss here we have 2 epochs. For epoch 0 the training loss is 11 and validation loss is 7, for epoch 1 the training loss is 7 and validation loss is 6.8.
- **Observation:** In Experiment-1, both training and validation losses decrease steadily indicating effective learning. In Experiment-2, while the initial training loss is higher, there is a significant reduction by the second epoch, suggesting that the model quickly adapts and improves its performance with additional training.

IoU Range	Area	MaxDets	AP
0.50:0.95	all	100	0.251
0.50	all	100	0.555
0.75	all	100	0.191
0.50:0.95	small	100	0.019
0.50:0.95	medium	100	0.226
0.50:0.95	large	100	0.451

(a) Table of AP



(b) AP for different IOU ranges and areas

Figure 4: Average Precision for exp-1

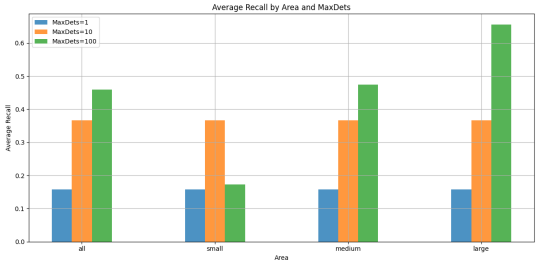
Figure 4: Average Precision Bar Graph(exp-1)
The bar graph shows the Average Precision (AP) for different IOU (Intersection over Union) ranges and areas.

- **(IOU 0.50, Area: all):** Yields the highest AP, indicating that the model performs best when using a lower IOU threshold across all object sizes.
- **Small areas (IOU 0.50:0.95, Area: small):** Have the lowest AP, suggesting that the model struggles with detecting smaller objects accurately.
- **Large areas (IOU 0.50:0.95, Area: large):** Show relatively high AP, indicating that the model is more effective at detecting larger objects across various IOU thresholds. Overall, the model performs better on larger objects and with a lower IOU threshold.

Figure 5: Average Recall for exp-1
The bar graph shows the impact of maximum detections and object size on Average Recall (AR)

IoU Range	Area	MaxDets	AR
0.50:0.95	all	1	0.158
0.50:0.95	all	10	0.366
0.50:0.95	all	100	0.459
0.50:0.95	small	100	0.173
0.50:0.95	medium	100	0.474
0.50:0.95	large	100	0.655

(a) Table of AR



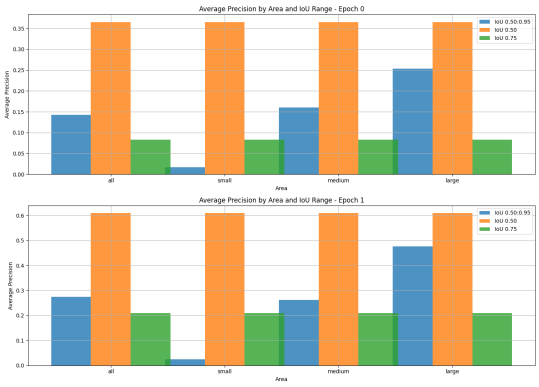
(b) Average Recall by Area and MaxDets

Figure 5: Average Recall for exp-1

- **MaxDets Impact:** AR improves with more detections: 0.158 (1), 0.366 (10), 0.459 (100).
- **Object Size Performance:** Small objects: AR is 0.173 (lowest). Medium objects: AR is 0.474. Large objects: AR is 0.655 (highest). Overall, the model performs better with larger objects and more detections, but struggles with small objects.

IoU	AP	Area	Epoch
0.50:0.95	0.143	all	Epoch 0
0.50	0.365	all	Epoch 0
0.75	0.083	all	Epoch 0
0.50:0.95	0.017	small	Epoch 0
0.50:0.95	0.160	medium	Epoch 0
0.50:0.95	0.253	large	Epoch 0
0.50:0.95	0.275	all	Epoch 1
0.50	0.609	all	Epoch 1
0.75	0.209	all	Epoch 1
0.50:0.95	0.024	small	Epoch 1
0.50:0.95	0.262	medium	Epoch 1
0.50:0.95	0.476	large	Epoch 1

(a) Table of AP



(b) Average Precision by area and IOU Range

Figure 6: Average Precision for exp-2

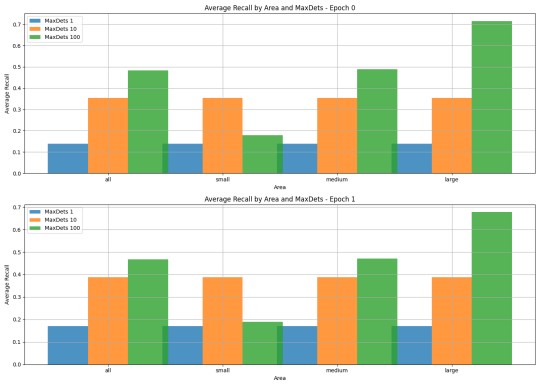
Figure 6: AP exp-2
The bar graph shows the Average Precision (AP) across epochs for different IOU Ranges and object sizes.

- **Epoch Progression:** Here we can see AP improves from Epoch 0 to Epoch 1, for IOU=0.50:0.95, AP increases from 0.143 to 0.275, for IOU=0.50, AP rises from 0.365 to 0.609, for IOU=0.75, AP improves from 0.083 to 0.209.
- **Object Size Performance:** Small objects: AP is low, among all. Increasing from 0.017 (epoch 0) to 0.024 (epoch 1). Medium objects: AP improves from 0.160 (epoch 0) to 0.262 (epoch 1). Large objects: AP increases significantly from 0.253 (epoch 0) to 0.476 (epoch 1). Overall The model's performance improves with more training epochs. The biggest gains in Average Precision (AP) are seen with larger objects. Small objects show only slight improvement, indicating the model performs better with larger objects and benefits from additional training.

Figure 7: AR exp-2
The bar graph shows the Average Recall (AR) across epochs for different IOU ranges, object sizes, and detection Limits.

IoU	AR	Area	MaxDets	Epoch
0.50:0.95	0.139	all	1	Epoch 0
0.50:0.95	0.354	all	10	Epoch 0
0.50:0.95	0.484	all	100	Epoch 0
0.50:0.95	0.179	small	100	Epoch 0
0.50:0.95	0.488	medium	100	Epoch 0
0.50:0.95	0.715	large	100	Epoch 0
0.50:0.95	0.171	all	1	Epoch 1
0.50:0.95	0.388	all	10	Epoch 1
0.50:0.95	0.468	all	100	Epoch 1
0.50:0.95	0.189	small	100	Epoch 1
0.50:0.95	0.471	medium	100	Epoch 1
0.50:0.95	0.677	large	100	Epoch 1

(a) Table of AP



(b) Average Recall by Area and MaxDets

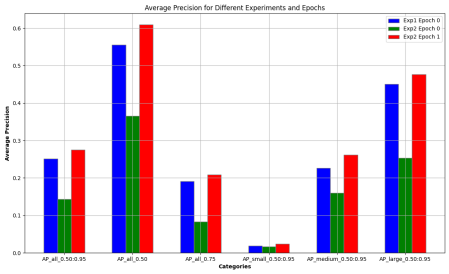
Figure 7: Average Recall for exp-2

- **Epoch Progression:** For IOU=0.50:0.95 and MaxDets=1, AR increases from 0.139 (epoch 0) to 0.171 (epoch 1), for IOU=0.50:0.95 and MaxDets=10, AR improves from 0.354 (epoch 0) to 0.388 (epoch 1), for IOU=0.50:0.95 and MaxDets=100, AR slightly decreases from 0.484 (epoch 0) to 0.468 (epoch 1).
- **Object Size Performance:** Small objects: AR is relatively low: 0.179 (epoch 0) and 0.189 (epoch 1). Medium objects: AR is moderate: 0.488 (epoch 0) and 0.471 (epoch 1), Large objects: AR is highest: 0.715 (epoch 0) and 0.677 (epoch 1). Overall The Average Recall (AR) of the model improves from epoch 0 to epoch 1, and the results are better for larger objects. Larger products show the most significant gains, while smaller products see only minor improvements. Overall, the model performs well on large objects and benefits from more training, although there are various effects on the increased detection limit.

Average Precision for Different Experiments and Epochs

Categories	Exp1 Epoch 0	Exp2 Epoch 0	Exp2 Epoch 1
AP_all_0.50:0.95	0.251	0.143	0.275
AP_all_0.50	0.555	0.365	0.609
AP_all_0.75	0.191	0.083	0.209
AP_small_0.50:0.95	0.019	0.017	0.024
AP_medium_0.50:0.95	0.226	0.16	0.262
AP_large_0.50:0.95	0.451	0.253	0.476

(a) Table of AP for both experiments



(b) AP for different experiments and epochs

Figure 8: AP for different experiments with different hyperparameters

Figure 8: Average Precision (AP) Metrics for Experiments with Different Hyperparameter Settings

This bar graph shows the comparison of Average Precision (AP) across different categories and IOU thresholds for two different experiment settings with varying hyperparameters, highlighting the performance improvements between experiments and epochs

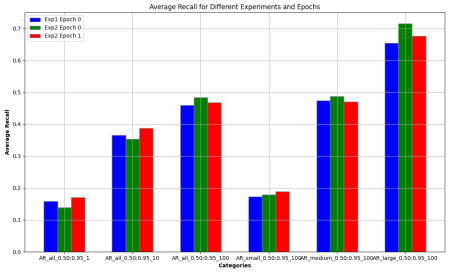
- **Epoch Improvement:** Exp 2 shows better results from epoch 0 to epoch 1, especially in AP_all_0.50, with improved performance due to additional training and changed hyperparameters.
- **Category Performance:** AP_all_0.50 is consistently the highest, while AP_small_0.50:0.95 remains the lowest, highlighting ongoing difficulty with small objects

- **Experiment Comparison:** Exp 1 (lr=0.0001, wd=0.0001) works better for medium objects and it works better when considering the overall accuracy across all object sizes at different IOU thresholds between 0.50 and 0.95. Exp 2 (lr=5e-5, wd=5e-5) improves at epoch 1, performing best for large objects and showing the highest accuracy at the 0.50 IOU threshold.
- **Summary:** Experiment 1 (lr=0.0001, weight decay=0.0001) we can see here it performs better for all categories. Experiment 2 (lr=5e-5, weight decay=5e-5) it shows a improvement at epoch 1, particularly at AP_all_0.50 and AP_lar_0.50:0.95 categories, where it does better than Experiment 1. Experiment 1 has better results for For medium and small objects, while Experiment 2 at epoch 1 performs better for larger objects and at the 0.5 IOU threshold.

Average Recall Metrics

Experiment	IoU	AR_all_1	AR_all_10	AR_all_100	AR_small	AR_medium	AR_large	IoU Range
Exp1	0.5	0.158	0.368	0.459	0.173	0.474	0.655	0.50:0.95
Exp2 Epoch 0	0.5	0.139	0.354	0.484	0.179	0.488	0.715	0.50:0.95
Exp2 Epoch 1	0.5	0.171	0.388	0.468	0.189	0.471	0.677	0.50:0.95

(a) Table of AR for both experiments



(b) AR for different experiments and epochs

Figure 9: AR for different experiments with different hyperparameters

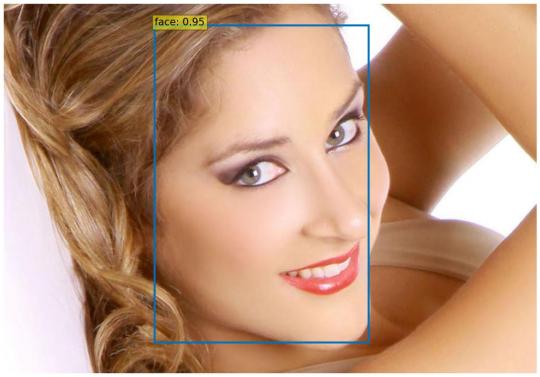
Figure 9: Average Recall (AR) Metrics for Experiments with Different Hyperparameter Settings

This bar graph shows the comparison of Average Recall(AR) across different categories and IOU thresholds for two different experiment settings with varying hyperparameters, highlighting the performance improvements between experiments and epochs.

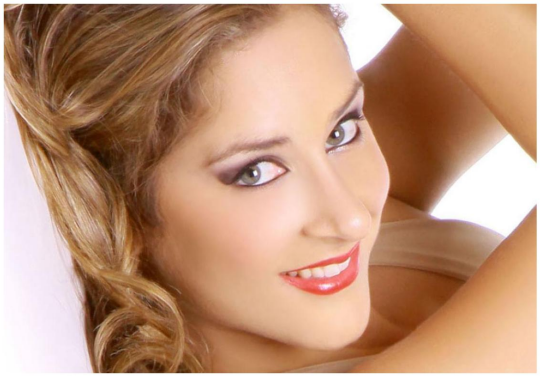
- **Epoch Improvement:** Exp 2, Epoch 1 (Red) shows an improvement compared to Exp 2, epoch 0 (Green) across most categories, particularly in AR_all_0.50:0.95_100 and AR_large_0.50:0.95_100, indicating better recall with more training.
- **Category Performance:**AR_large_0.50:0.95_100 consistently has the highest recall across all experiments, suggesting that larger objects are easier to detect accurately. AR_small_0.50:0.95_100 consistently has lower recall, which highlights the difficulty in detecting small objects.
- **Experiment Comparison:** Exp 1, epoch 0 (Blue) generally performs better in medium and overall categories but shows slightly lower recall in large objects. Exp 2, epoch 0 (Green) and epoch 1 (Red) improve upon Exp 1 in some large object categories, particularly after additional training at epoch 1.
- **Summary:** Experiment 1 has a strong performance across most categories, especially for medium objects. Experiment 2 shows significant improvements in large object detection by epoch 1, indicating that fine-tuning and additional training benefit these specific categories.

Figure 10: Detection Results on Test Images

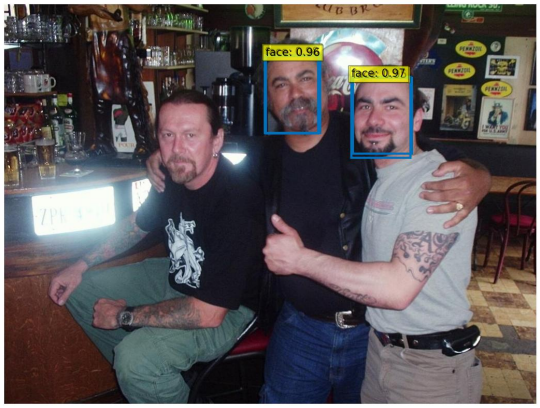
Here are the test images showing the results for the model with different hyperparameters in each experiment.



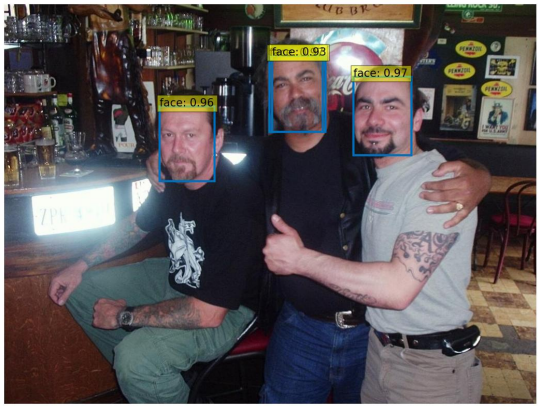
(a) Exp-1



(b) Exp-2



(c) Exp-1



(d) Exp-2

Figure 10: Detection on test images for both experiments

List of Abbreviations

DETR Detection Transformer

IOU Intersection over Union

AP Average precision

AR Average Recall

FP False positive

FN False Negative

TP True positive

mAP Mean Average precision

mAR Mean Average Recall

NMS NonMaxSuppression

LR Learning rate

GPU Graphics Processing Unit

CNN Convolutional Neural Network

Bbox Bounding Box

Epoch Training Cycle

MSE Mean Squared Error

ResNet Residual Network

ReLU Rectified Linear Unit

1 Introduction and Background

This project focuses on training the DETR model on custom-dataset for object detection, DETR was introduced by facebookAI(2020) they uses transformer architecture to detect and classify objects in image with end-to-end training. Here we evaluate the performance of DETR on custom dataset with changing the hyperparameters to enhance model accuracy, and analyze the effectiveness of the model in detecting and classifying objects. Doing DETR on custom dataset helps in handling various detection tasks demonstrating the model's flexibility. These results can help to improve real-world applications in detection. Object detection is a important task in computer vision it has two parts object localization, object detection where localization finds a single object in an image, detections finds multiple objects in an image.In past there were some techniques called sliding window for detection this has problems like it take a lot of computation and many bounding boxes for same object. So here comes deep learning which has greatly improved both accuracy and reliability. DETR (Detection Transformer) is a groundbreaking model to detect items, It uses a method based on transformers. DETR differs from older models, It sees object detection as a problem of predicting sets, It uses transformers to understand the whole image. The dataset we made for this project looks a lot like the COCO dataset. It has many different kinds of pictures with various object types and settings. We picked this dataset to see how well DETR works in a setup that's close to COCO.

2 Problem Statement

How does the DETR (Detection Transformer) model's performance for a custom dataset depend on changes in hyperparameters like learning rate and weight decay? In two independent trials, precision and recall metrics are used to assess the performance of the model in order to determine how these hyperparameter adjustments affect the model's detection accuracy.

3 Current Approach

- **Dataset Preparation:** For this project the wider face dataset is used (refer fig-2) and was converted into the COCO format to create a custom dataset for training and evaluating the DETR model. Here we extract the image paths and bounding box annotations, converting like the coco format. Each image was annotated with face bounding boxes and additional metadata like image dimensions and category IDs. JSON files were generated both for training and validation data, guaranteeing DETR compatibility. Only photos with a reasonable amount of face annotations were added to the dataset, preserving its quality and usefulness for efficient model testing and training.
- **Model Configuration:** Here the DETR model was setup with pre-trained weights from link . By starting with pre-trained weights, i trained the model on custom dataset, which was based on the Wider face dataset, more efficiently. The model was made specifically for the face detection task by using a single object category("face") and reining the category ID and other related settings to the coco format used in the dataset.

- **Hyperparameter Tuning:** In order to customize the DETR model for the specific dataset, hyperparameter tuning was done by means of two different experiments(refer fig-8 and fig-9). Learning rate and weight decay were two of the key hyperparameters that were tuned, both of which were essential for the training of the model. The purpose of these experiments was to determine the best settings that would improve the model's precision and recall in detecting faces. This project aimed to vary systematically the hyperparameters in order to maximize the model's accuracy.
- **Performance Evaluation:** The DETR model was tested on a custom dataset and metrics precision and recall were used to analyze the model's performance. These metrics provide insights into the model's ability to detect and classify the faces(refer fig-4,fig-5,fig-6 and fig-7). The model was trained with different hyperparameter settings, especially by varying the learning rate and weight decay and the evaluation was done by comparing the results of these two experiments.
- **Analysis and Comparison:** The project examined how well the DETR model performed in two tests with various hyperparameters. I observed how changes in learning rate and weight decay affected face detection accuracy by measuring precision and recall for each configuration. In order to provide recommendations for the ideal model setup on the custom dataset, this comparison showed which hyperparameter values offered the optimum balance between properly detecting faces and minimizing errors.

4 Contributions

- **Creating a Custom Dataset:** I chose the dataset which is specifically for the face detection task(wider-face). This dataset is used for training and validating the model, which would help in getting desired result.
- **Adapting the DETR Model:** I customized the DETR my specific needs which i trained it on my dataset and demonstrated how DETR can be adjusted to work well for different types of object detection tasks.
- **Evaluating Model Performance:** For two experiments i evaluated metrics called precision and recall, which helped me to understand the performace of the model.
- **Analyzing Hyperparameter Impact:** I analyzed how different hyperparameters impact the performace of model in two experimets, so that we can optimize the model by giving the correct hyperparameter values.
- **Providing Optimization Recommendations:** I did 2 experimets by changing the hyperparameters and i gave the settings of hyperparameters(refer fig-8 and fig-9) for which the model performs better which would help in future research.

5 Learnings

- **Object detection:** Object detection is a computer vision task, that uses to locate objects in an image which has the bounding box around the object and a class label is given to the object to understand that this object is detected. This technique is very useful in autonomous vehicles(tesla car),and image search.
- **Transformer:** In DETR transformer plays a crucial rule(Detection transformer), firstly the image is divided into patches, these patches are send to CNN to extract features from the image like eye, colors etc. There will a technique called positional encoding

where each patch belongs in the original image. This is like giving each patch a label that tells the model it's exact position in the image. These positional encoded patches are then sent to the transformer encoder these encoder processes the features to capture the relationship's between different parts of the image using self-attention mechanism. Further these encoded parts are sent to transformer decoder to make predictions, the encoder understands the image, and the decoder uses this understanding to figure out what objects are present and where they are located. The decoder output's a fixed number of prediction's.

- **Neural networks:** Neural networks consist of different layers. The first layer is the input layer, the second is the various hidden layers, and finally, the output layer. Initially, the input layer consists of all the features that need to be passed to the neural network to learn. Neural networks use the backpropagation technique to reduce the loss. The weights are updated using this formula:

$$w_{\text{new}} = w_{\text{old}} - h\frac{\partial L}{\partial w_{\text{old}}}$$

  Here, the features and weights are passed, and the output is predicted using the forward propagation method. The loss function used is mean squared error, given by the formula $L = (y - \hat{y})^2$. Here, y is the actual output and $\hat{y}$ is the predicted output. Here backpropagation is used as optimizer's, optimizer's are used to reduce the loss.

- **Hyperparameter tuning:** If the model performs well on the training set but poorly on validation set, it might be overfitting. Hyperparameter's such as learning rate, no.of layers, weight-decay can be adjusted to improve performance. The goal is to find the best hyperparameter's that make the model perform well on both the training and validation sets. There are effectively 49 convolutional layer's within the ResNet-50. These layer's are responsible for extracting feature's from the input image. There are total 96 attention head's. These attention head's are used as part of the transformer's architecture to process and understand the relationship's between different parts of the input data.

- **ResNet-50:** Here the ResNet-50 is used as backbone, the backbone is used to extract features from the input image, these feature's are then utilized by the rest of the model, particularly the transformer encoder-decoder architecture to make the final object detection prediction's.

- **Precision:** Ratio of correctly predicted positive observations to the total predicted positives.

$$\text{Precision} = \frac{\text{True Positive (TP)}}{\text{True Positive (TP)} + \text{False Positive (FP)}}$$

  AP measures how well the model predicts the bounding boxes.

- **Recall:** The true positive rate (TPR), or the proportion of all actual positives that were classified correctly as positives, is also known as recall.

$$\text{Precision} = \frac{\text{True Positive (TP)}}{\text{True Positive (TP)} + \text{False Negative(FN)}}$$

  AR measures how many of the actual object's are correctly detected by model.

- **IOU(Intersection over union:** Intersection over Union (IoU) is a measure that shows how well the prediction bounding box aligns with the ground truth box.

$$\text{IOU} = \frac{\text{Area of Intersection}}{\text{Area of Union}}$$

  If IOU > 0.5, this says that the overlap(that is the bounding box we got while detecting) is decent.

6 Summary

In this project i aimed to enhance the DETR for object detection using different experiments by changing hyperparameters, firstly i took the dataset Wider face where my work is to detect faces in image, i formatted that dataset annotations into coco format because i used the pre-trained model which is already trained on coco dataset that is ResNet-50, i used this backbone and trained on my dataset. I conducted two experimets to see how hyperparameters impact the performance of DETR model. For experimet-1 i took the hyperparameters as lr and weight decay and did training, then to see the performance evaluation for experiment-1 i used the metrics called precision and recall. With same hyperparameters just by changing the values i performed experiment-2 and got the metrics. After i compared both experiments results i got these results finally(refer-fig 4,fig-5,fig-6,fig-7,fig-8 and fig-9) for AR experiment-1 has a strong performance across most categories, especially for medium objects. Experiment-2 shows significant improvements in large object detection by epoch 1, indicating that fine-tuning and additional training benefit these specific categories. And for AP experiment-1 (lr=0.0001, weight decay=0.0001) generally performs better for most of the categories. Experiment-2 (lr=5e-5, weight decay=5e-5) shows a large improvement at epoch 1, particularly at AP_all_0.50 and AP_lar_0.50:0.95 categories, where it does better than experiment-1. Experiment-1 has better results for For medium and small objects, while experiment-2 at epoch 1 performs better for larger objects and at the 0.5 IOU threshold.

7 Future work:

I would like to increase the number of epochs for each experiment where i did for 1 epoch in experiment-1 and 2 epochs for experiment-2, also DETR doesn't uses convolutional filters which are in CNN, there are responsible for extracting features like, shapes, edges from the input images, where DETR uses only transformers which doesn't have these filters. In future i need to improve and address this areas.

References

[1] Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A. and Zagoruyko, S., 2020. End-to-End Object Detection with Transformers. *European Conference on Computer Vision (ECCV)*, 12346, pp.213-229.

[2] Zhu, X., Su, W., Lu, L., Li, B., Wang, X. and Dai, J., 2020. Deformable DETR: Deformable Transformers for End-to-End Object Detection. *International Conference on Learning Representations (ICLR).*

[3] Misra, I., Zitnick, C.L., Hebert, M. and Girshick, R., 2021. End-to-End Object Detection with Grid-Based Representations. *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp.8300-8309.

[4] Dai, Z., Cai, B., Lin, Y. and Chen, J., 2021. Dynamic DETR: End-to-End Object Detection With Dynamic Attention. *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp.2988-2997.

[5] Meng, D., Chen, X., Fan, Z., Zeng, G., Li, H., Yuan, Y., Sun, L. and Wang, J., 2021. Conditional DETR for Fast Training Convergence. *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp.3651-3660.