Multispectral Object Detection using DETR

UNIVERSITY of HOUSTON | ECE

Alan Devkota, Jayson Varughese, Utkarsh Gupta, Yidan Shen

Table of Contents

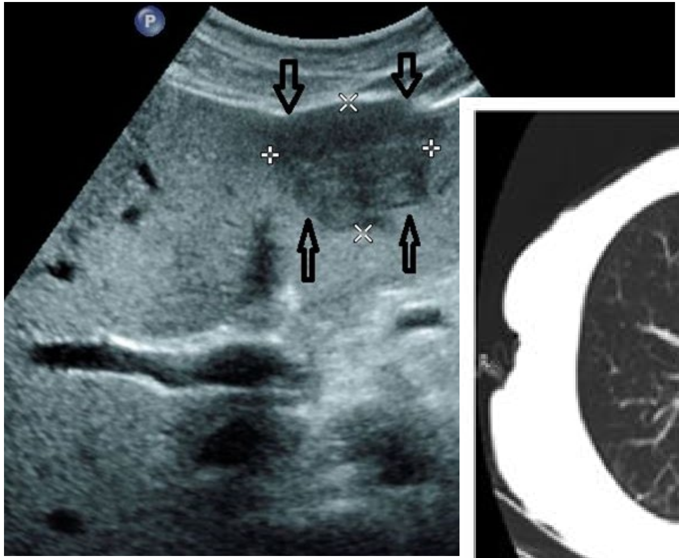
- Introduction
    - Application
    - Related work
    - Dataset
    - Challenges
- Transformer Architecture
- Methodology
- Results
- Future Work
- References

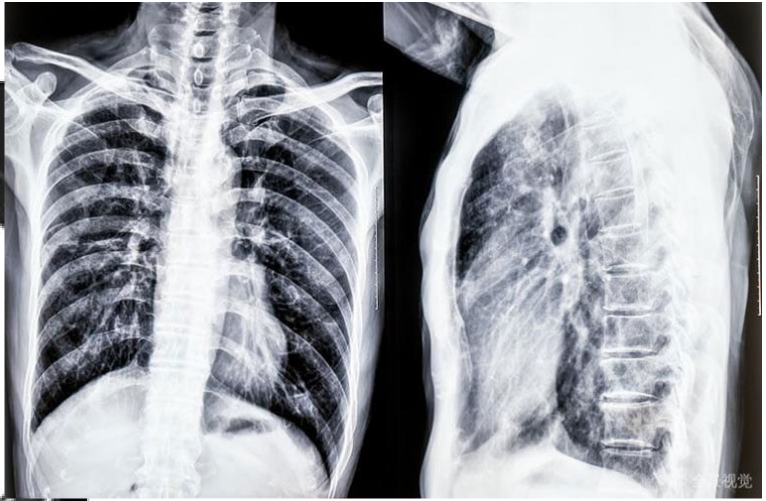
Introduction

- Multispectral data for object recognition and location
- Cross-attention
- Detection Transformers (DETR)

Application

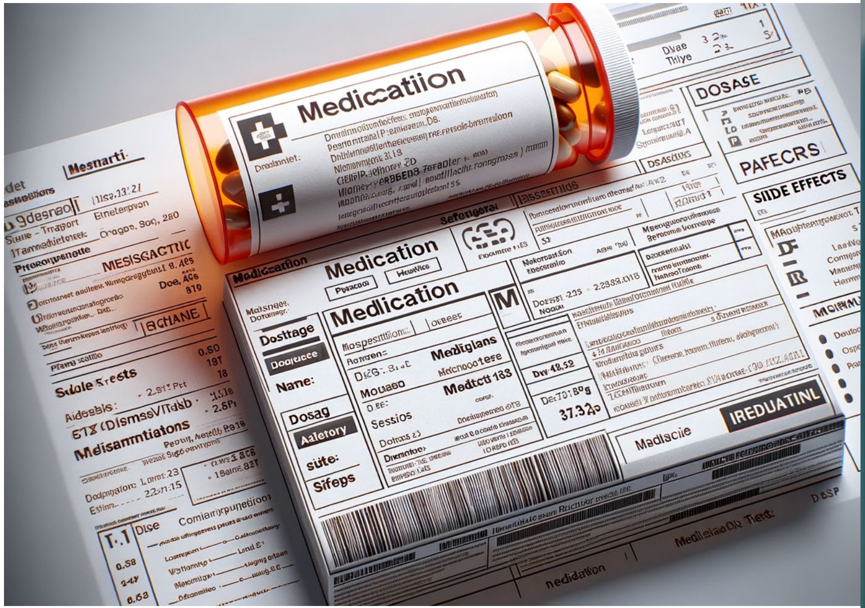
- Healthcare– patient diagnosis (UltraSound, CT, X-rays)





Application

- Medication (text based modality), test, daily routine dataset (Tabular text)



Application

- Autonomous driving (camera, IR, radar)

Application

- Military reconnaissance (camera, IR, night-vision equipment)

Related work

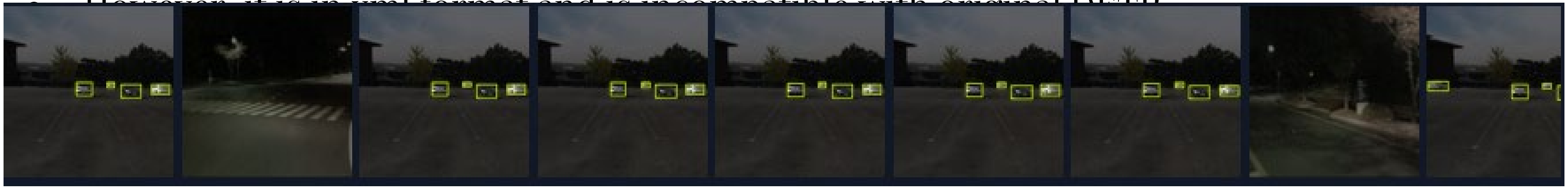
End-to-End Object Detection with Transformers

- Object detection method using transformer architecture, achieving an end-to-end process.
- Simplifies object detection by directly predicting without needing region proposals or complex feature extraction.
- Utilizes the transformer's self-attention mechanism to better adapt to different object sizes, shapes, and understand global context.
- Implements bipartite matching algorithm in training to streamline model-ground truth object associations.

Dataset

KAIST Dataset:

- Multispectral pedestrian detection dataset.
- Developed by Korea Advanced Institute of Science and Technology.
- Includes visible light and infrared images.
- Captured during various times, weather, and lighting conditions.
- Utilized for advanced computer vision algorithms.
- Applicable in autonomous driving and urban surveillance.
- Aids in pedestrian identification and tracking.
- Addresses complex environments and diverse spectral characteristics.
- However, it is in xml format and is incompatible with original DETR

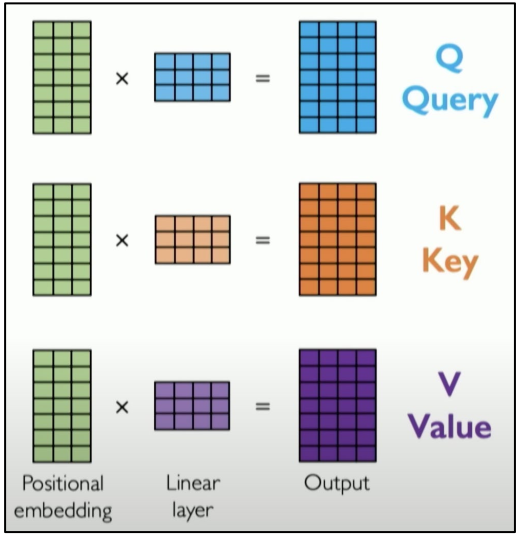


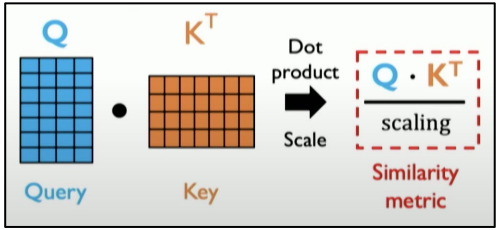
Challenges

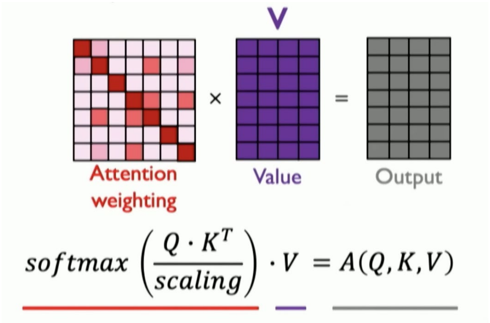
- KAIST dataset annotations differ from DETR's COCO format.
- Original DETR model requires modification for the encoder.
- Determining layer count and attention in encoders.
- Hard to find fully aligned multimodal datasets, unannotated ones need pre-processing.
- Training on small datasets yields poor results, large datasets needed.
- Uses DETR architecture with 2 parallel modalities, requiring token concatenation in encoders/decoders.
- Matching tensor dimensions in each layer for multimodal dataset is challenging.

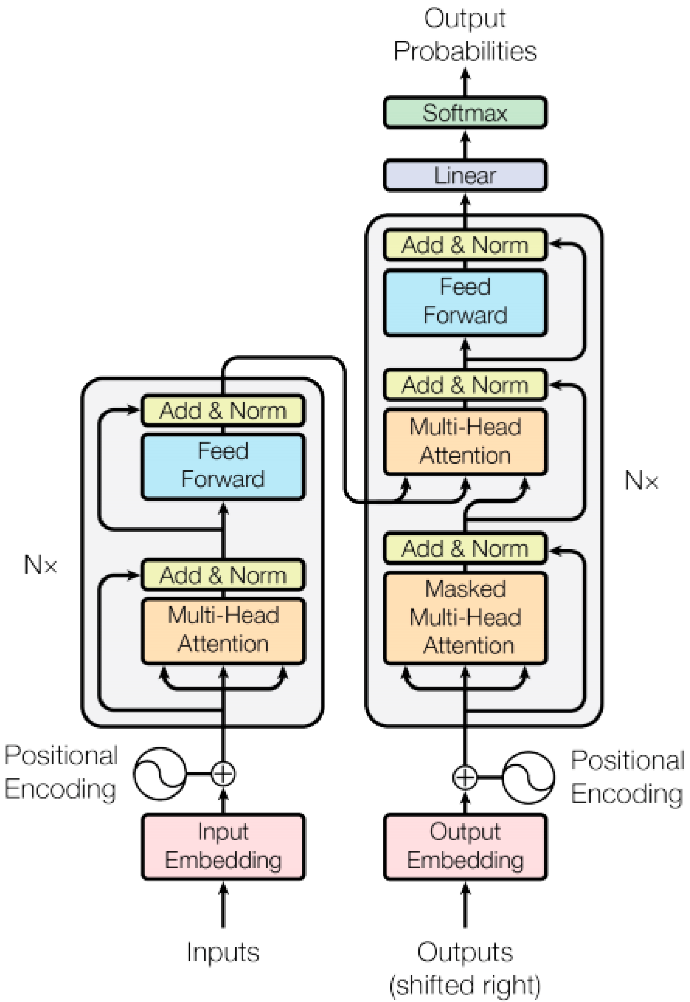
Attention is all you need

Self Attention Mechanism used in transformer







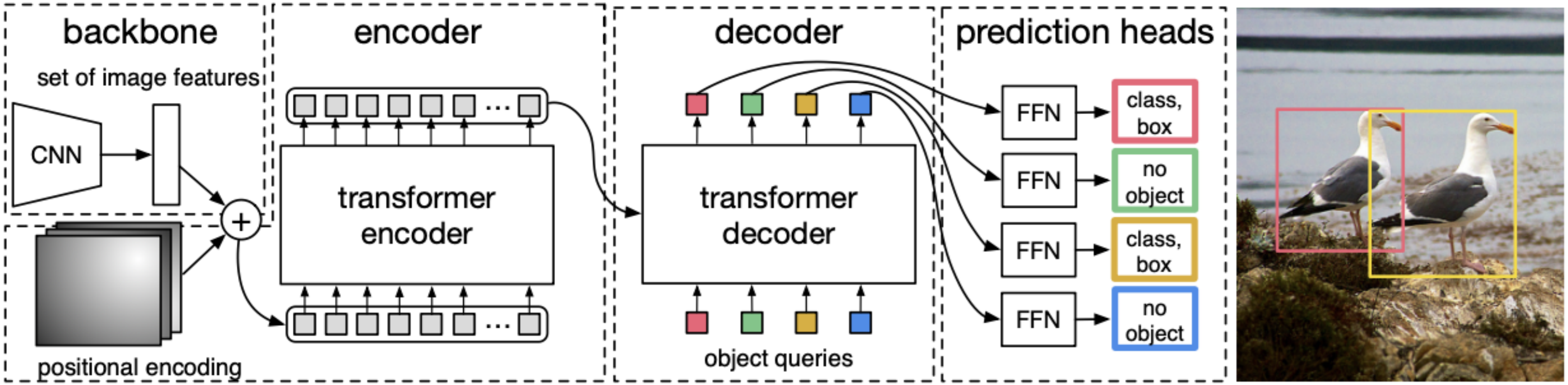


DETR Architecture

CNN Backbone

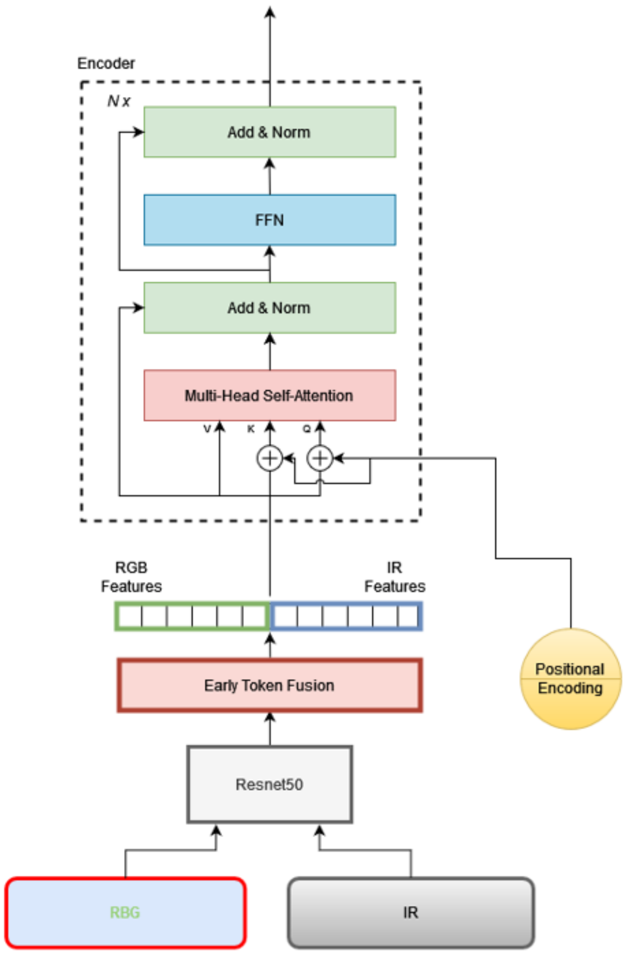
Encoder - Decoder transformers

Feed forward network (FFN)



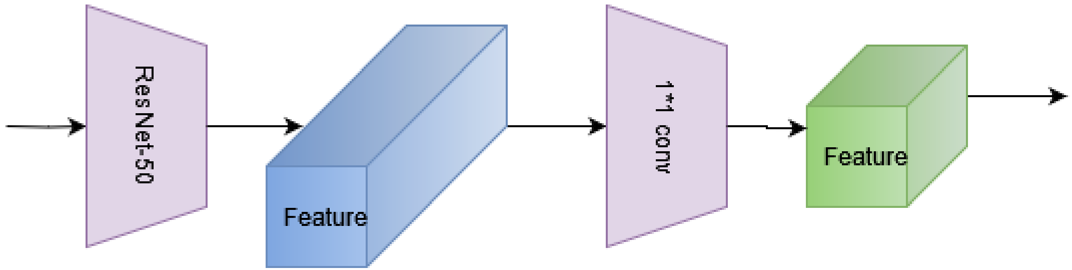
Methodology (1st stage)

- Two modalities
  - RGB
  - IR
- Feature extract
  - Resnet-50
  - FFN
- Encoder
  - Multi-head Self-attention
  - FFN
- Positional Encoding
  - DetrSinePositionalEmbedding



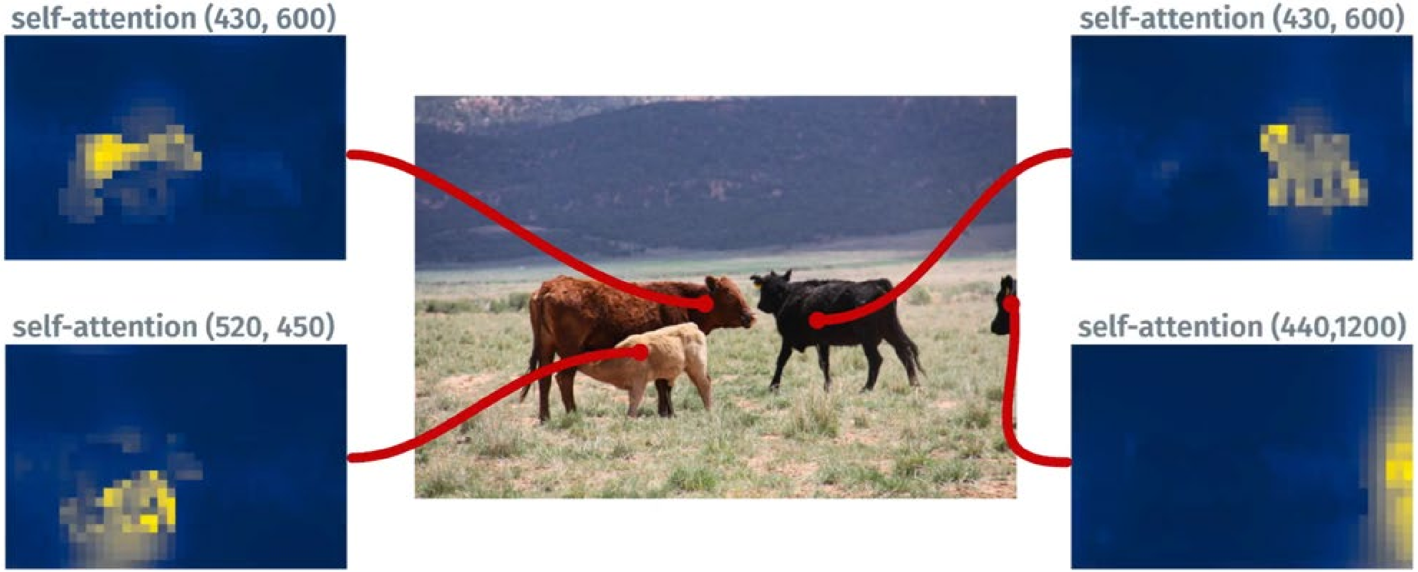
Methodology (1st stage)

- ResNet-50



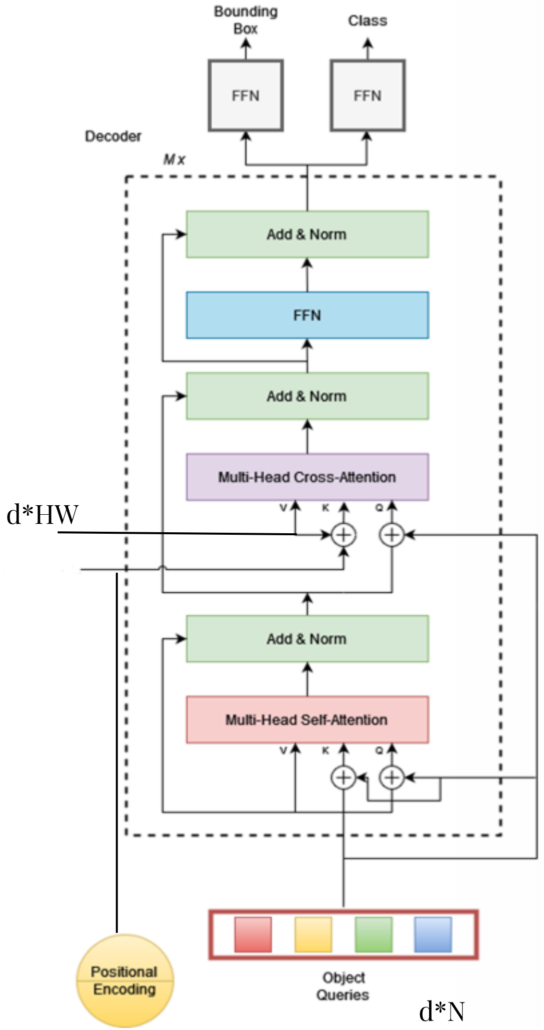
Methodology (1st stage)

- Role of encoder
  - Pixel belonging to same image have high attention
  - Repeat for other source points



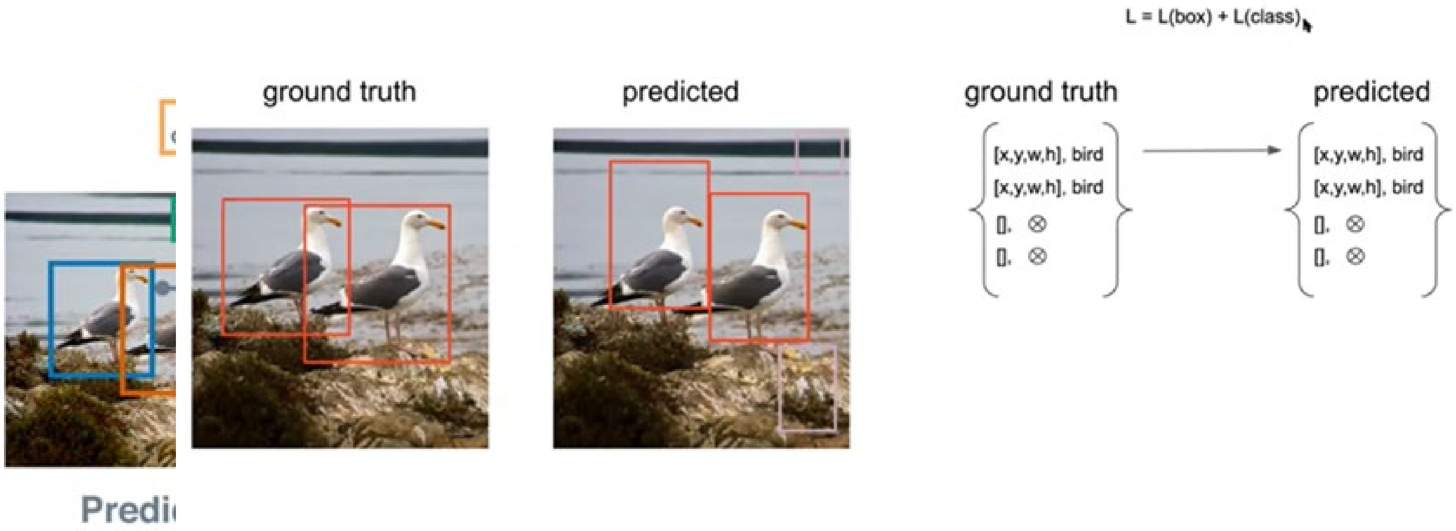
Methodology (2nd stage)

- Object queries
- Decoder
    - Multi-head Self-attention
    - Multi-head Cross-attention
    - FFN
- FFN
    - Bounding box
    - Class
- Positional Encoding
    - DetrSinePositionalEmbedding

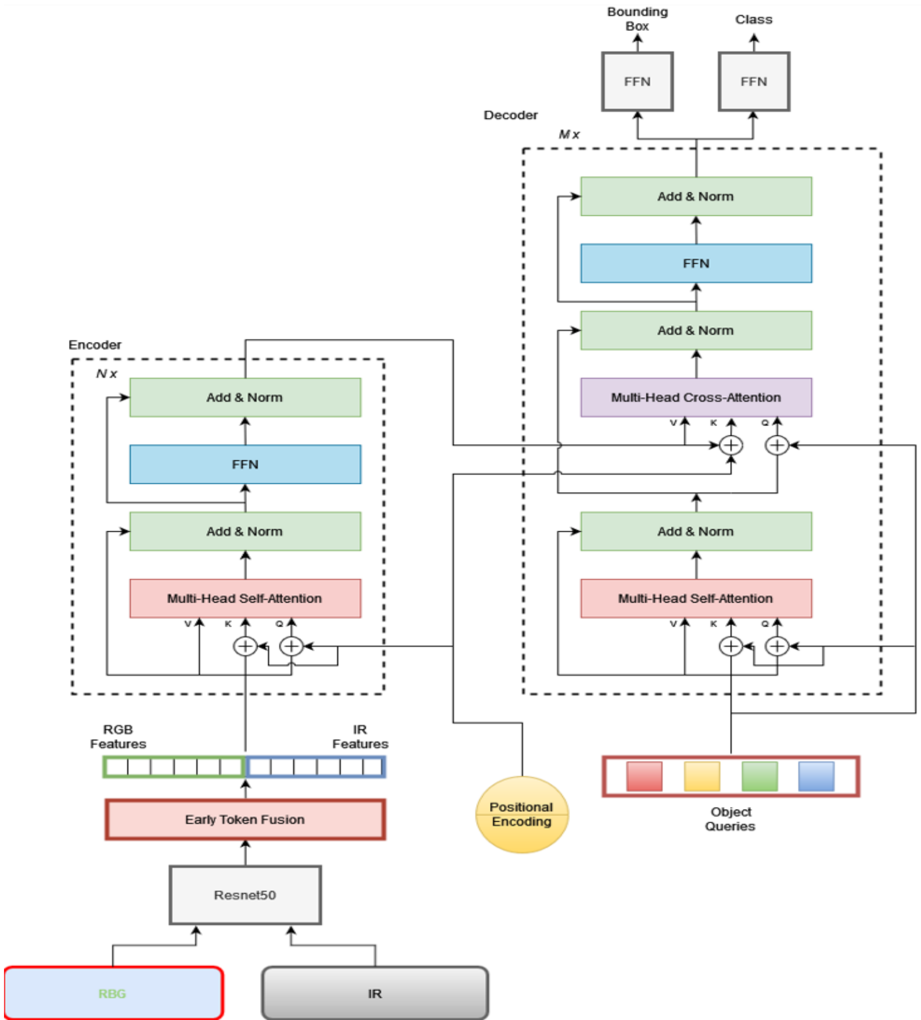


Methodology (2nd stage)

- Bipartite Matching
  - Hungarian algorithm



Methodology



Environment

```
| NVIDIA-SMI 515.86.01    Driver Version: 515.86.01    CUDA Version: 11.8     |
|-------------------------------+----------------------+----------------------+
| GPU  Name        Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf  Pwr:Usage/Cap|         Memory-Usage | GPU-Util  Compute M. |
|                               |                      |               MIG M. |
|===============================+======================+======================|
|   0  Tesla V100-DGXS...  Off  | 00000000:07:00.0  On |                    0 |
| N/A   38C    P0    38W / 300W |    190MiB / 32768MiB |      0%      Default |
|                               |                      |                  N/A |
+-------------------------------+----------------------+----------------------+
|   1  Tesla V100-DGXS...  Off  | 00000000:08:00.0 Off |                    0 |
| N/A   36C    P0    39W / 300W |      9MiB / 32768MiB |      0%      Default |
|                               |                      |                  N/A |
+-------------------------------+----------------------+----------------------+
|   2  Tesla V100-DGXS...  Off  | 00000000:0E:00.0 Off |                    0 |
| N/A   37C    P0    39W / 300W |      9MiB / 32768MiB |      0%      Default |
|                               |                      |                  N/A |
+-------------------------------+----------------------+----------------------+
|   3  Tesla V100-DGXS...  Off  | 00000000:0F:00.0 Off |                    0 |
| N/A   35C    P0    37W / 300W |      9MiB / 32768MiB |      0%      Default |
|                               |                      |                  N/A |
+-------------------------------+----------------------+----------------------+


+-----------------------------------------------------------------------------+
| Processes:                                                                  |
|  GPU   GI   CI        PID   Type   Process name                  GPU Memory |
|        ID   ID                                                   Usage      |
```

Settings

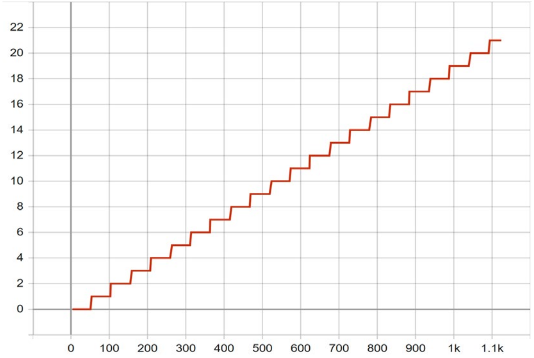
- Input image:
  512x640
- Resnet 50, 6 x DETR encoder layers, 6 x DETR Decoder layers, two classifier layers.

Settings:

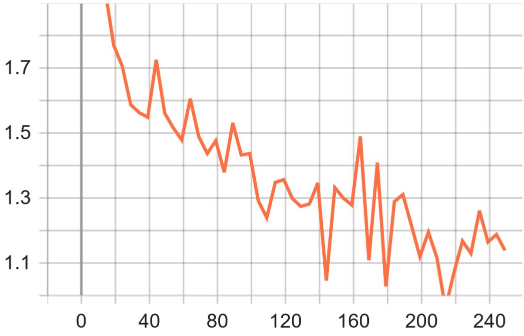
- Number of epoch for training = 50, logs will be written every 5 steps.
- Trained on Tesla V100
- Initial learning rate = 1*10^-4 for main part, 1*10^-5 for backbone, weight decay used for regularization = 1*10^-5 (learning rate scheduler would change these rates)

- Total 41.5 M parameters, 41.3 M are trainable , 222K are non-trainable, Total estimated size of model parameters = 116.01 MB

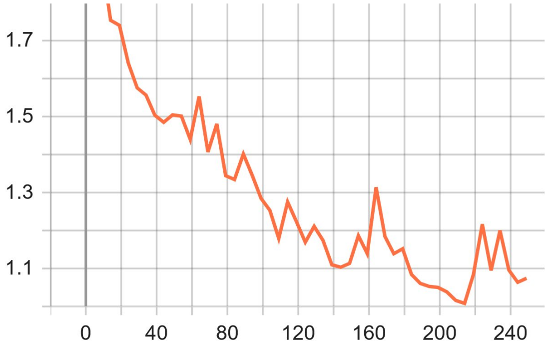
Training Results



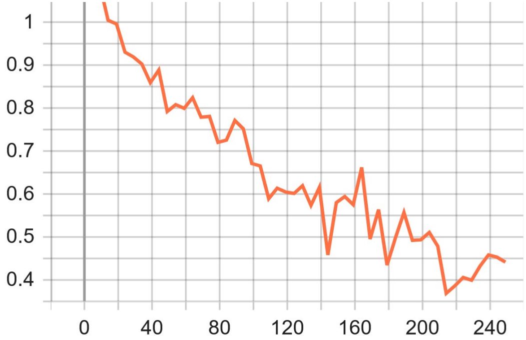
epoch



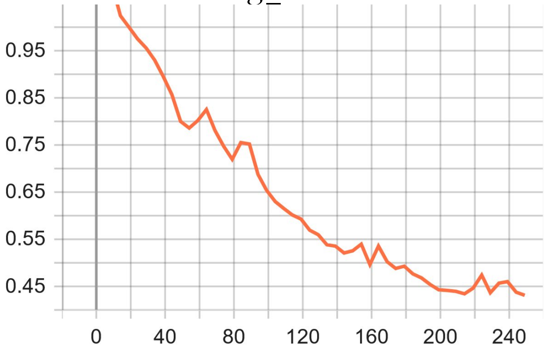
training_loss



validation_loss

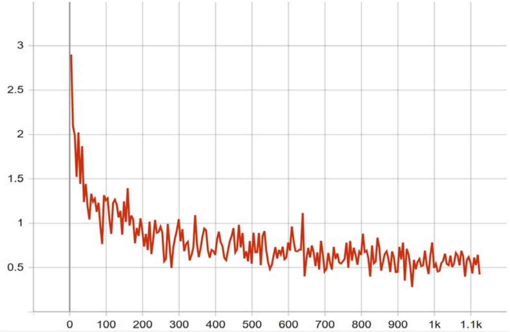


train_loss_ce

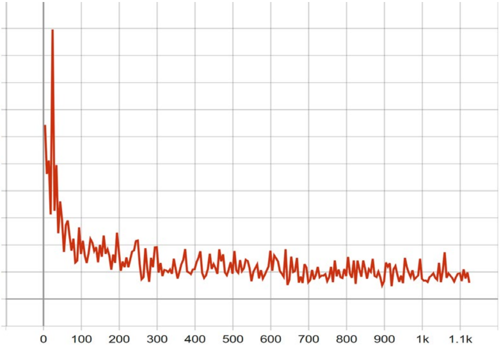


validation_loss_ce

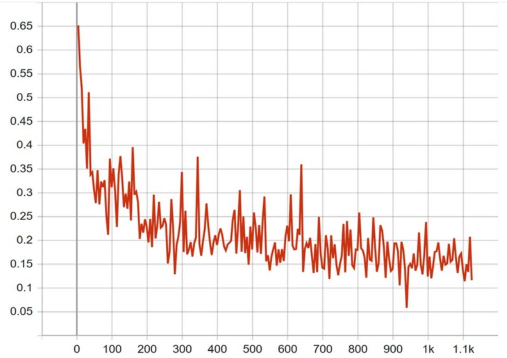
Results



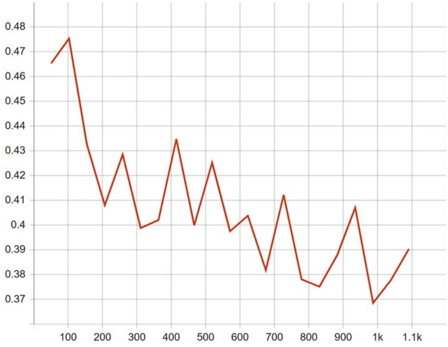
training_loss



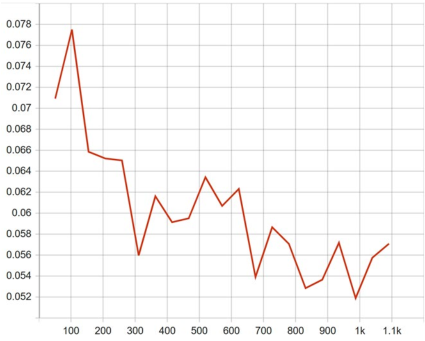
train_loss_bbox



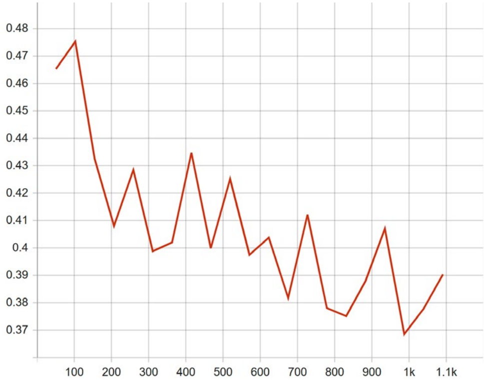
train_loss_giou



validation_loss

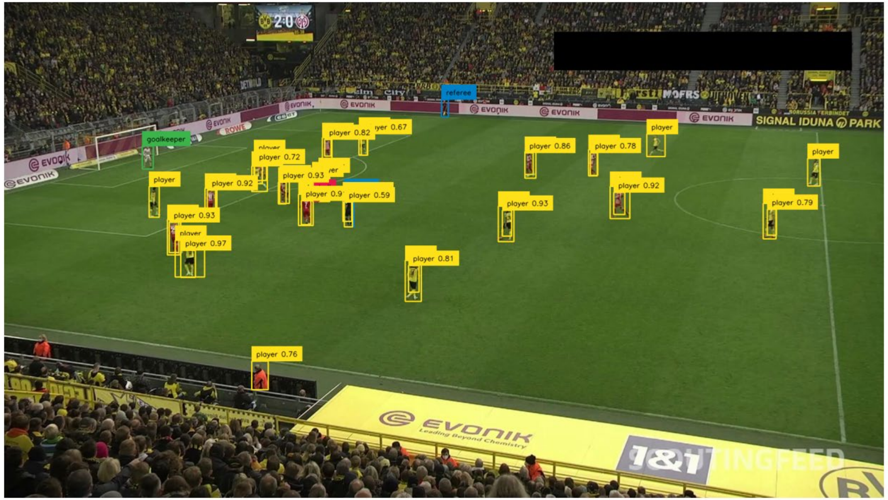


validation_loss_bbox

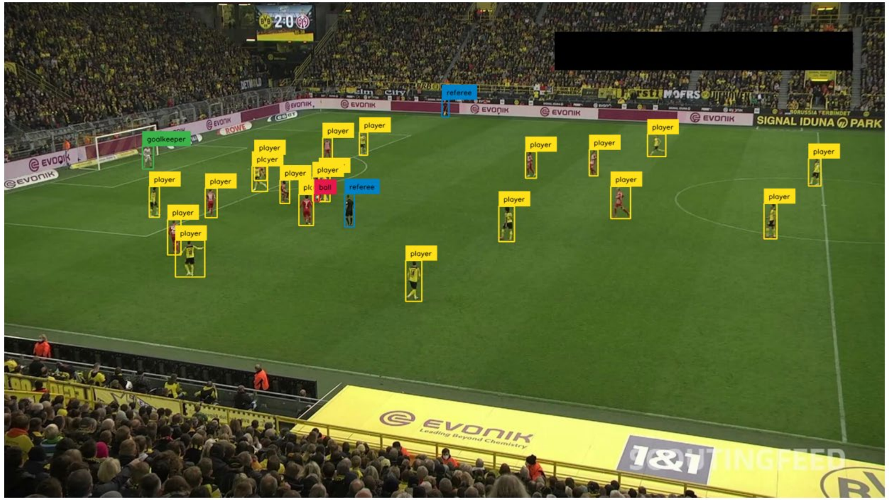


validation_loss_giou

Detection Examples for FootballPlayers Dataset

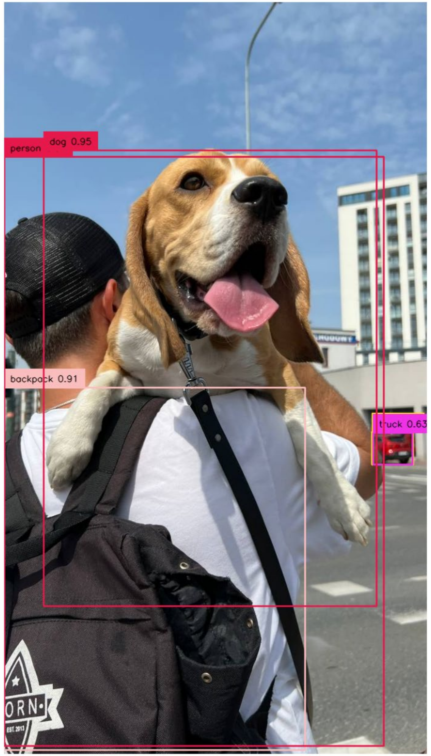


image_comp_actual_detected

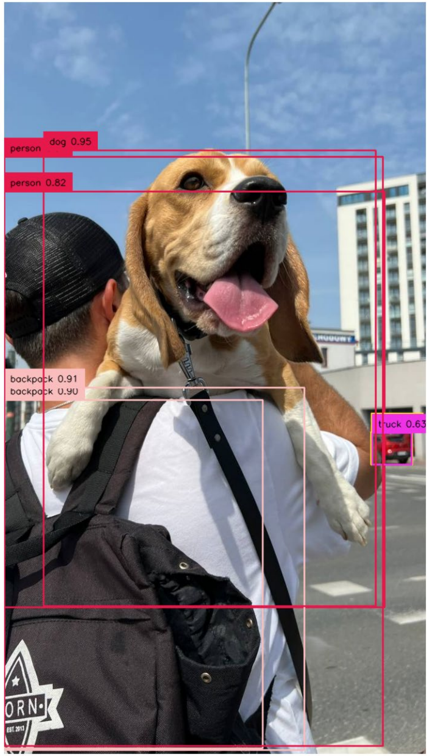


image_compared_to_detection

Detection Example for Random Object (with and without NMS)



Initial with NMS



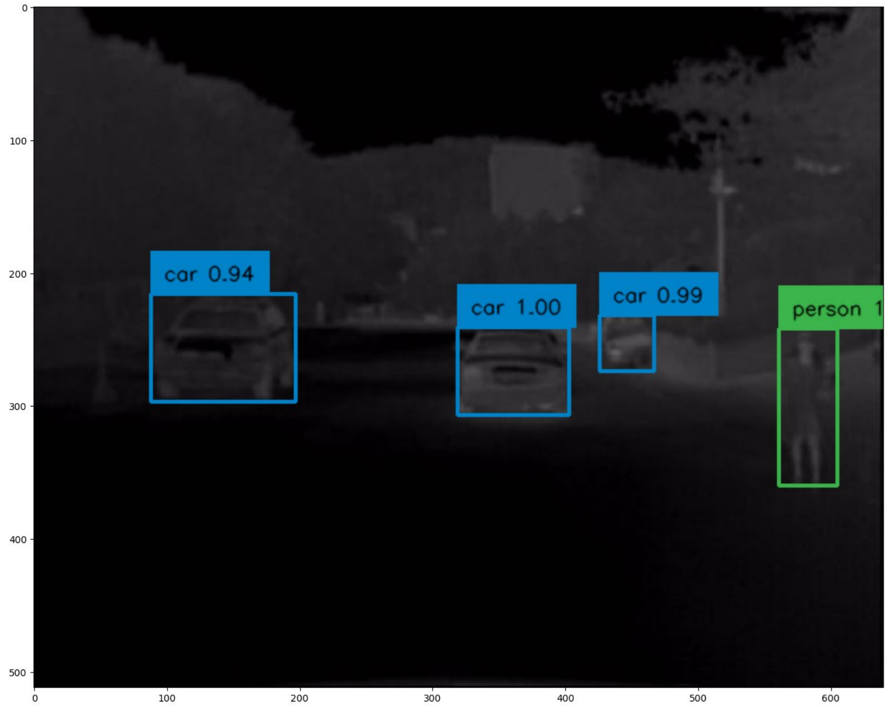
Initial without NMS

Example of KAIST dataset

IR Image

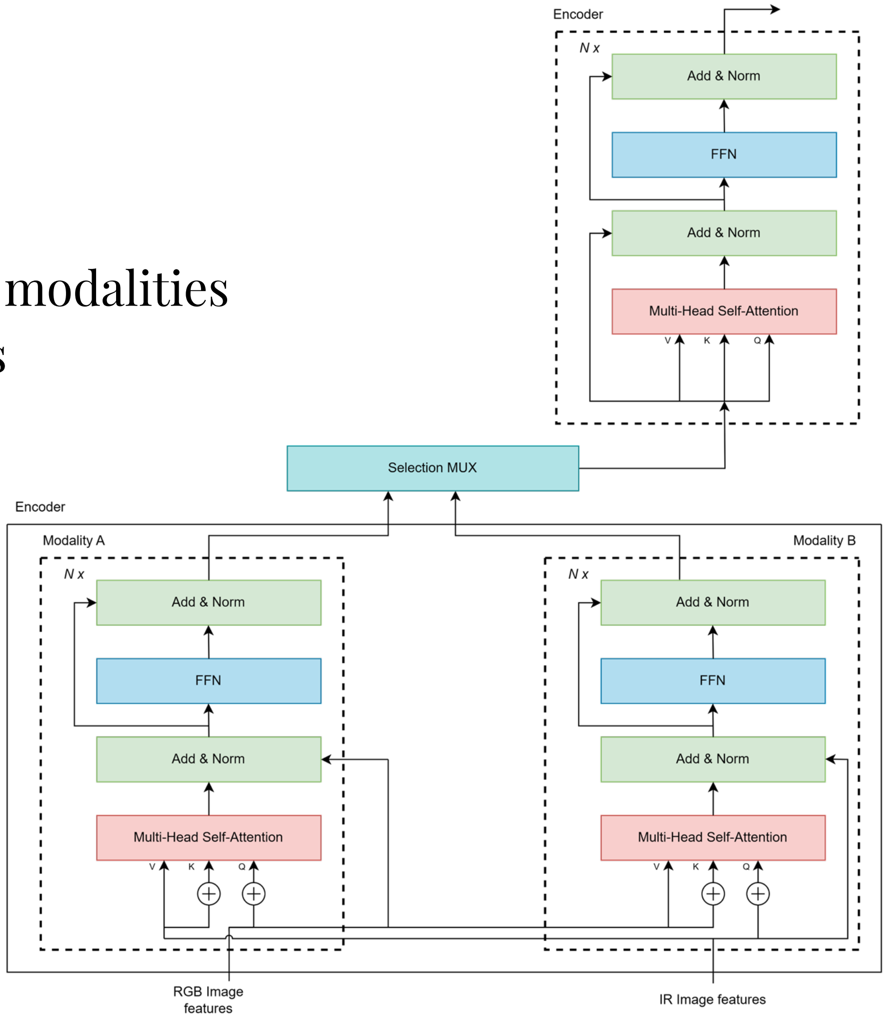
RGB Image

Results on KAIST Dataset



Future Work

- Extend the two modalities to three modalities
- New architecture use two encoders



References

[1]. Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., & Zagoruyko, S. (2020). End-to-End Object Detection with Transformers. arXiv preprint arXiv:2005.12872.
[2]. Cai, Z., Vasconcelos, N.: Cascade R-CNN: High quality object detection and instance segmentation. PAMI (2019)
[3]. Glorot, X., Bengio, Y.: Understanding the difficulty of training deep feedforward neural networks. In: AISTATS (2010)
[4]. He, K., Girshick, R., Doll´ar, P.: Rethinking imagenet pre-training. In: ICCV (2019)
[5]. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR (2016)
[6]. Hosang, J.H., Benenson, R., Schiele, B.: Learning non-maximum suppression. In: CVPR (2017)
[7]. Hu, H., Gu, J., Zhang, Z., Dai, J., Wei, Y.: Relation networks for object detection. In: CVPR (2018)
[8]. Kirillov, A., Girshick, R., He, K., Doll´ar, P.: Panoptic feature pyramid networks. In: CVPR (2019)
[9]. Kuhn, H.W.: The hungarian method for the assignment problem (1955)
[10]. Lin, T.Y., Goyal, P., Girshick, R.B., He, K., Doll´ar, P.: Focal loss for dense object detection. In: ICCV (2017)
[11]. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Doll´ar, P., Zitnick, C.L.: Microsoft COCO: Common objects in context. In: ECCV (2014)
[12]. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., Chintala, S.: Pytorch: An imperative style, high-performance deep learning library. In: NeurIPS (2019)
[13]. Ren, S., He, K., Girshick, R.B., Sun, J.: Faster R-CNN: Towards real-time object detection with region proposal networks. PAMI (2015)
[14]. Salvador, A., Bellver, M., Baradad, M., Marqu´es, F., Torres, J., Gir´o, X.: Recurrent neural networks for semantic instance segmentation. arXiv:1712.00617 (2017)
[15]. Tian, Z., Shen, C., Chen, H., He, T.: FCOS: Fully convolutional one-stage object detection. In: ICCV (2019)
[16]. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention is all you need. In: NeurIPS (2017)
[17]. Zhou, X., Wang, D., Kr¨ahenb¨uhl, P.: Objects as points. arXiv:1904.07850 (2019)

Thank You

Miscellaneous

```
  | Name  | Type                   | Params
-------------------------------------------------
0 | model | DetrForObjectDetection | 41.5 M
-------------------------------------------------
41.3 M    Trainable params
222 K     Non-trainable params
41.5 M    Total params
166.010   Total estimated model params size (MB)
```

- Input image:
  512x640
- Resnet 50, 6 x DETR encoder layers, 6 x DETR Decoder layers, two classifier layers.

- **Settings:**
- Number of epoch for training = 50, logs will be written every 5 steps.
- Trained on Tesla V100
- Initial learning rate = 1*10^-4 for main part, 1*10^-5 for backbone, weight decay used for regularization = 1*10^-5 (learning rate scheduler would change these rates)