Multispectral Object Detection using DETR

Alan Devkota, Jayson Varughese, Utkarsh Gupta, Yidan Shen
University of Houston

I. INTRODUCTION

Multispectral object detection involves identifying and pinpointing objects in a scene using data from various spectral bands or wavelengths. In contrast to standard RGB imaging, multispectral data captures a broader range of the electromagnetic spectrum, including non-visible wavelengths like infrared. This extended spectral information enhances the precision of object detection across diverse applications.

The methodology of multispectral object detection utilizes data from multiple spectral bands, with sensors often incorporating infrared bands such as near-infrared (NIR) and shortwave infrared (SWIR) alongside the visible spectrum. This capability allows for the detection of subtle environmental variations, making it particularly valuable in fields like precision agriculture for monitoring crop health, environmental monitoring for land cover changes, and military applications for reconnaissance and target detection.

While integrating multispectral data presents challenges such as increased data complexity and computational demands, the potential benefits, including improved accuracy and richer information, make multispectral object detection a valuable and continually advancing field. This is especially true in remote sensing applications where satellites and aircraft capture multispectral data for Earth observation and analysis.

A topic we cover in this experimentation is using Detection Transformers (DETR) as a means to conduct Multispectral Object Detection. DETR is a state-of-the-art object detection model in computer vision. Unlike traditional object detection methods that rely on region proposal networks and anchor boxes[13,15,2], DETR uses a transformer architecture, which was originally developed for natural language processing tasks. In DETR, the image is divided into a fixed number of regions, and each region is treated as a separate "object query." The transformer processes these queries in parallel, attending to the entire image simultaneously, a method contrasting with learnable NMS methods and relation networks that explicitly model relations between different predictions with attention [7]. This approach eliminates the need for separate region proposals and feature extraction steps.

DETR's key innovation is its direct set-based prediction approach. Instead of predicting bounding boxes and class scores independently, DETR predicts all object queries and their corresponding positions and classes in a single forward pass. The model uses a bipartite matching mechanism to associate predicted queries with ground truth objects, allowing it to handle object detection as a set prediction problem[14]. This set-based paradigm enables DETR to achieve impressive results in terms of accuracy and efficiency. It has shown effectiveness in various object detection benchmarks and has become a notable model in the field of computer vision. DETR's design showcases the versatility of transformer architectures beyond natural language processing, illustrating their effectiveness in handling complex visual tasks.

Our focus of research is to develop a transformer model that integrates the information from different modalities together to enhance the prediction as well as address the challenges posed by missing modalities. Our modified DETR transformer encoders harness the vast potential of heterogeneous data via early fusion of modalities features (RGB and IR) unlike the original DETR transformer model. Afterward, the model performs object detection by using a DETR decoder with object queries.

II. RELATED WORK

The article produced by Facebook AI regarding Object Detection using Transformers[1] introduces a paradigm-shifting methodology for object detection by leveraging transformer architectures, similar to those used in sequence prediction [16]. Departing from the traditional two-stage approach, this model streamlines the entire process into a single end-to-end framework. The core innovation lies in the direct prediction of object detections without the need for explicit region proposals and extensive feature extraction, differing from methods relying on large sets of proposals [15,2], anchors [10], or window centers [17,16].

The transformer's self-attention mechanism proves instrumental in processing images divided into a set of learnable object queries[4]. Unlike traditional methods relying on predefined anchor boxes, the model adapts dynamically to different object scales and shapes. The capacity of the transformer to capture long-range

dependencies is harnessed for understanding global context, enabling a more comprehensive perception of the entire scene.

A significant advancement is the incorporation of a bipartite matching algorithm during training[9]. This mechanism facilitates the direct association of predicted queries with ground truth objects, simplifying the training process and eliminating the need for post-processing steps. The end result is a model capable of simultaneously predicting object classes and bounding boxes in a single pass.

Evaluation on benchmark datasets like COCO [11] underscores the model's competitive performance compared to conventional two-stage detection approaches, such as Faster R-CNN [13]. The end-to-end design, coupled with the inherent strengths of transformers, suggests a promising direction for the evolution of object detection methods. Beyond achieving state-of-the-art results, the paper contributes to the broader exploration of transformer architectures in visual tasks, pushing the boundaries of what is achievable in object detection and paving the way for more efficient and effective computer vision systems.

The DETR architecture is characterized by its simplicity and is composed of three main components: a CNN backbone for feature extraction[5], an encoder-decoder transformer, and a straightforward feed-forward network (FFN) for the final detection prediction[6]. Unlike many contemporary detectors, DETR can be implemented in various deep learning frameworks with minimal lines of code[12]. The backbone starts with an initial image, extracting features through a conventional CNN to produce a lower-resolution activation map.

Positional encodings are added to account for the transformer's permutation invariance[4]. Then, after reducing the channel dimension of the activation map using a 1x1 convolution, and generating a new feature map[3], the transformer encoder is applied. This map is then treated as a sequence, and each encoder layer employs a multi-head self-attention module and a feed-forward network. The transformer decoder follows a standard architecture but decodes N objects in parallel at each layer, in contrast to the autoregressive model used by Vaswani et al[16]. The N object queries, serving as positional encodings, are transformed into output embeddings and independently decoded into box coordinates and class labels.

The final prediction is computed by a 3-layer perceptron with ReLU activation and a linear projection layer. This predicts the normalized box coordinates and class labels using a softmax function. To handle a fixed-size set of N bounding boxes, an additional special class label ∅ is introduced to represent cases where no object is detected within a slot[6].

Auxiliary decoding losses, including prediction FFNs and Hungarian loss, are employed during training to assist the model in outputting the correct number of objects for each class[8]. These losses are applied after each decoder layer, and shared layer normalization is used to normalize inputs to the prediction FFNs from different decoder layers.

Detection Transformers are introduced as a novel object detection system built on transformer architecture and bipartite matching loss for direct set prediction. In evaluations on the challenging COCO dataset, DETR achieves results comparable to an optimized Faster R-CNN baseline[13]. Its implementation is straightforward, and its flexible architecture easily extends to panoptic segmentation, delivering competitive performance[11]. Notably, DETR outperforms Faster R-CNN on large objects, attributed to its ability to process global information through self-attention[11].

While DETR presents promising results, it also introduces new challenges, particularly in training, optimization, and performance on small objects. The paper acknowledges that addressing these challenges might require further work, drawing parallels with the evolution of current detectors that needed several years of refinement to overcome similar issues. The authors anticipate future research efforts to successfully tackle these challenges and further enhance the capabilities of DETR[11].

III. EXPERIMENTAL METHODOLOGY

Our methodology leverages a dual-modality approach using both RGB and Infrared (IR) images to enhance the robustness and accuracy of object detection. Both modalities assume a distinct role which provides a more holistic understanding of the object(s) of focus. The process is subdivided into three main stages, a backbone of CNN to extract the features from both RGB and Thermal IR images, a transformer architecture that consists of an encoder and decoder to learn the contextual information between the embedded features, and two classifier units to predict class and bounding box for each object detected inside an image. Here we are using ResNet50 to extract features of both RGB and thermal images and then provide early token fusion by concatenating the extracted features from ResNe50 together

and computing attention between the tokens of RGB and IR modalities to get learned feature representations. Our approach relies on transformer architectures, which are powerful tools that are typically used for language tasks. The versatility of these transformers allows us to extend behavior and comprehension of the larger-scale environment.

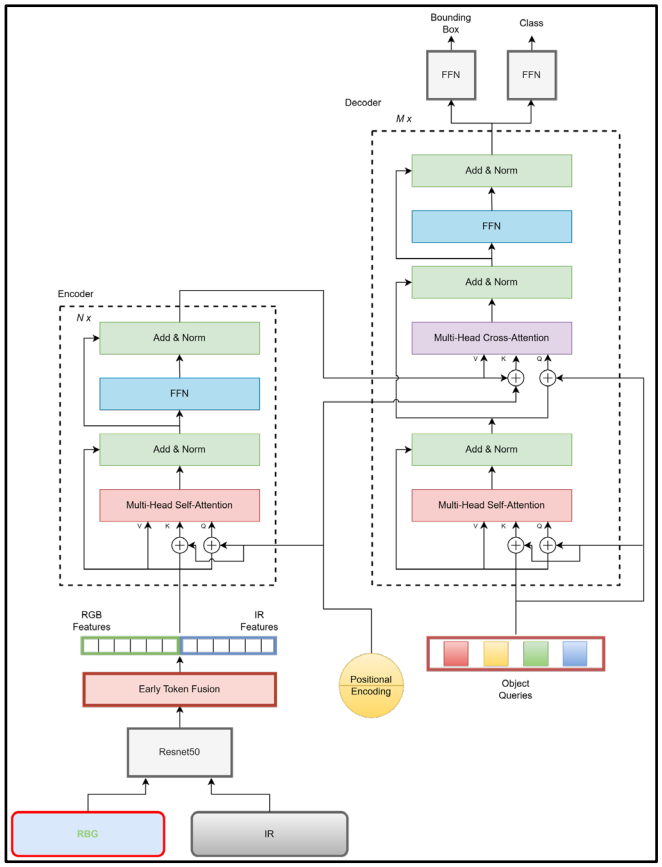

Figure 1: Overall Transformer Architecture

A. Early Fusion of RGB and IR Features

The initial step involves tokenizing the RGB and IR images from the KAIST dataset by breaking down the visual information into manageable units that can be analyzed by the model. The feature representations are then extracted using a pre-trained Resnet50 model. The features extracted are then concatenated together to perform an early fusion. These concatenated features are passed to the original DETR encoder, thus the embedding dimension would be double that of the original model. During the attention calculation, these features serve as the queries *(Q)*, keys *(K)*, and values *(V)* in the transformer's encoder. A multi-headed self-attention mechanism is applied, allowing the model to focus on relevant parts of the image as well as on the similarity or closeness of RGB and Thermal pairs.

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right)V$$

$$\text{where } Q = YW^Q, K = XW^K, V = XW^V$$

Equation 1: Attention Equation

Here, W is a trainable weight matrix, initialized as random in the beginning and used to form Q, K, and V. This attention mechanism on concatenated features allows the model to discern intricate details and relationships between different aspects of the image between two modalities, enhancing its ability to accurately detect objects from the provided dataset. Thus, the attention mechanism is crucial in enabling the model to weigh the importance of different visual elements that span across the image as it can extract missing features from one modality and also enhance the learned representation by combining the modalities.

Following this attention phase in the encoder, a fixed set of learnable encodings called object queries would attend with the keys and values output from the encoder via cross-attention. Therefore, object queries would detect corresponding objects in the output features from the encoder. A multi-layer perceptron (MLP) block is used to process these attention-guided features. The MLP block is a sophisticated component designed to process the features that have been refined through the cross-attention mechanism. The MLP block acts as a computational mechanism that applies non-linear transformations to the extracted features, further accentuating the similar and distinct patterns from the image dataset to provide precise object detection.

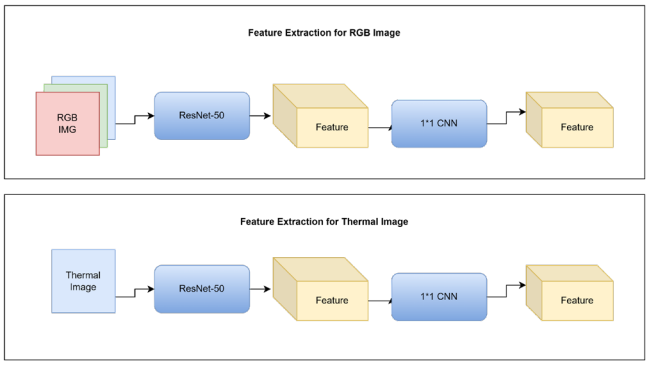

Figure 2: First Stage RGB/IR Processing

B. Set Prediction Loss

The amalgamated features after concatenation after the cross-attention between object queries and the key and values from the encoder output are then utilized for two

concurrent objectives: classification and object localization. Classification is achieved through a predictive model, which employs the cross-entropy loss function to ascertain the probability distribution over predefined classes. For object localization, a bounding box regression model delineates the spatial coordinates of the object instances within the input data. The performance of the bounding box regression is quantified through a loss function suitable for continuous output values.

Adopting the DETR model, instead of using the anchor boxes and processing the entire set of objects in an image, we introduce a set-based paradigm for object detection. The key functions used for this process are Classification Loss, Bipartite Matching loss, No Object Loss and Bounding Box loss. The classification loss is computed as cross-entropy loss. Let *Pij* is the predicted probability of the j-th class and *yi* is the ground truth for the class label of the i-th object. The equation for classification loss and bipartite match is given as:

$$\mathcal{L}_{\text{class}} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{C} y_{ij}\log(p_{ij}),$$

Equation 2: Classification Loss Equation

where *N* is the number of objects, C is the number of classes.

The Bipartite machine loss uses Hungarian algorithm to associate predicted objects with ground truth objects to find the optimal assignment matrix. If *M* is the assignment matrix, then the value of *Mij* is 1 when the i-th predicted object is matched with the j-th ground truth object, and 0 otherwise. The Bipartite loss is defined as:

$$\mathcal{L}_{\text{match}} = \frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{N} M_{ij}\mathcal{L}_{\text{box}}(b_{ij},\hat{b}_{ij}) + \lambda\mathcal{L}_{\text{class}}(c_{ij},\hat{c}_{ij}),$$

Equation 3: Bipartite Loss Equation

where b_{ij} and $\hat{b}_{ij}$ are the predicted coordinates and the ground truth bounding box. λ_{box} is a smooth L1 for bounding box coordinates, and λ is a balancing power. The No Object Loss is defined as:

Equation 4: No Object Loss Equation

where c_{ij} and $\hat{c}_{ij}$ are the class probabilities for the i-th predicted objects and ground truth. The bounding box loss measures the difference between the predicted and ground truth bounding box is computed via smooth L1 loss.

Equation 5: Bounding Box Loss Equation

Thus, total loss is:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{class}} + \lambda_{\text{match}}\mathcal{L}_{\text{match}} + \lambda_{\text{no\_obj}}\mathcal{L}_{\text{no\_obj}} + \lambda_{\text{box}}\mathcal{L}_{\text{box}},$$

Equation 6: Total Loss Equation

where λ_{match}, $\lambda_{\text{no\_obj}}$ and λ_{box} are weighting parameters whose values are typically determined through experiment. Also, the Intersection over Union (IoU) is more commonly used as

$$\mathcal{L}_{\text{no\_obj}} = \frac{1}{N}\sum_{i=1}^{N}(1 - \sum_{j=1}^{N_{\text{obj}}} M_{ij})\mathcal{L}_{\text{class}}(c_{ij},\hat{c}_{ij}),$$

an evaluation metric to assess how well the predicted bounding box aligns with the ground truth bounding boxes. It helps to evaluate the performance of our model.

Prior to processing the images, it was vital to understand the annotations associated with the provided

$$\mathcal{L}_{\text{box}}(b_{ij},\hat{b}_{ij}) = \sum_{k\in\{x,y,w,h\}} \text{SmoothL1}(b_{ij}^{k} - \hat{b}_{ij}^{k})$$

dataset. Standard KAIST dataset provided annotations in a format different than what would be adaptable to our use cases regarding processing of images. The KAIST dataset natively used annotations provided in XML format, which would not be an ideal fit for our object detection and image processing architectures. Python code was developed to handle the conversion of the KAIST dataset from XML format to COCO format, which allowed for Dataset Sanitization. The main goal of Dataset Sanitization is to ensure that the data is accurate, reliable, and free from errors or inconsistencies prior to feeding into the model, which allows for meaningful analysis and training to take place.

IV. RESULTS

We trained our model on a Tesla V100-DGXS GPU. The input images had a dimension of 512x640 which was scaled to 512x512 before feeding to our model. The different components that were used were ResNet50 with 50 layers, DETR encoder layers, 6 DETR Decoder layers, and two linear layers for the bounding box and classifier.

The number of epochs for training was 22, and we recorded logs every five steps. The initial learning rate used was 1*10^-4 for the main transformer part and 1*10^-5 for the ResNet backbone. Weight decay was used for regularization with the value of 1*10^-5. We implemented a learning rate scheduler that would update this rate. Overall, we had 41.5M parameters, 41.3M of which were trainable,

and 222K of which were non-trainable. The total estimated size of the model parameter was 116.01MB.

Even though the multispectral KAIST dataset was converted from the XML format into COCO format with limited annotation, our model achieves comparable results with the SOTA DETR transformer model. We performed longer training to decrease the loss functions and be stable, so that better predictions would be achieved on the KAIST dataset. Comparing the training loss which is composed of the classification loss, bounding box loss, and GIoU loss in **(Fig. 3 and Fig. 4)**, our model is achieving significantly better results than the KAIST dataset, with about 1.25 times improvement. Next, when comparing the validation loss in **(Fig. 5 and Fig. 6)**, we see that our validation loss is way less than the one with the DETR model. However, the validation loss is not stable due to the low quality of our dataset. Next, we compare the training loss of our model with the DETR model as shown in **(Fig. 7 and Fig. 8)**. Our training loss converges very well and provides superior performance for the KAIST dataset as shown in **(Fig. 9 and Fig. 10)**. When comparing the validation loss, our model begins to overfit after reaching 800 iterations, but the validation error is also lower compared to the DETR on the KAIST dataset. The results would have been better if we stopped after 800 iterations.

Moreover, we have also plotted the training loss for bounding box, validation loss for bounding box and training loss and validation loss for intersection of union in our model to measure our performance.

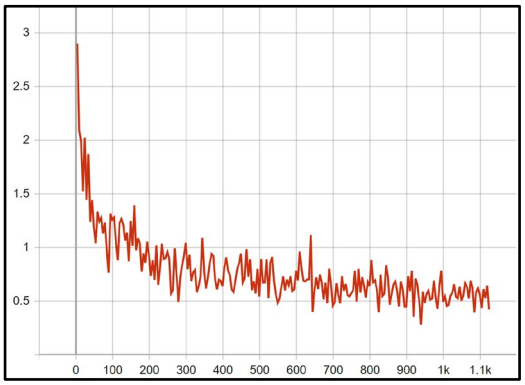

Figure 3: Training loss of our model on KAIST

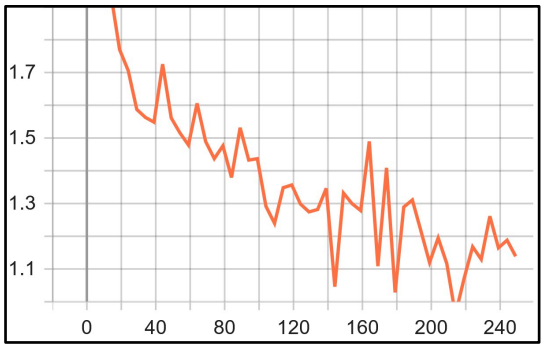

Figure 4: Training loss of DETR model on KAIST

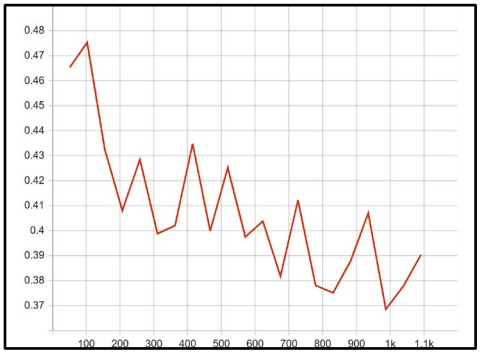

Figure 5: Validation loss of our model in KAIST

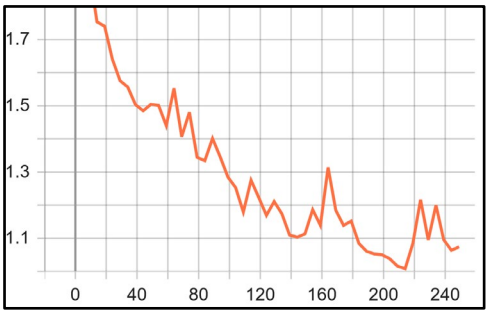

Figure 6: Validation loss of DETR model on KAIST

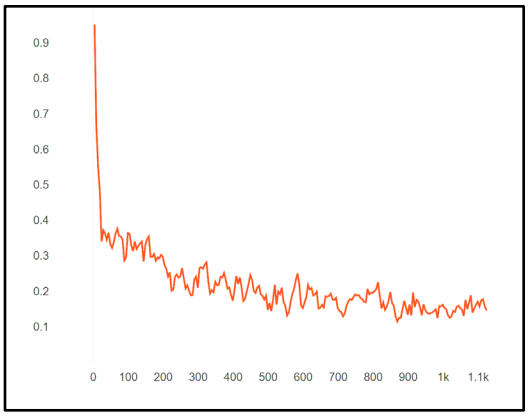

Figure 7: Train loss for classification of our model on KAIST

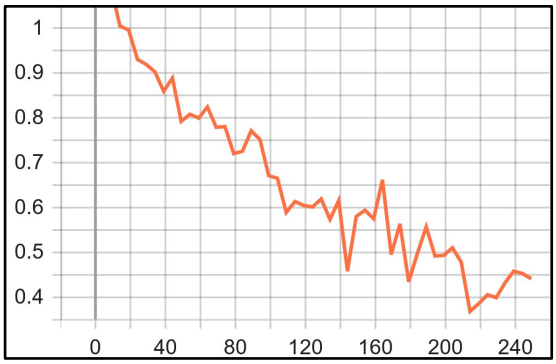

Figure 8: Train loss for classification of DETR on KAIST

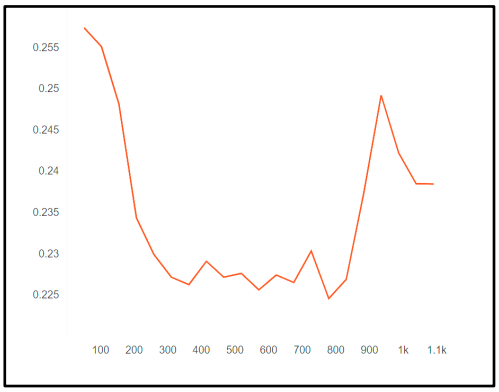

Figure 9: Validation loss of our model for classification on KAIST

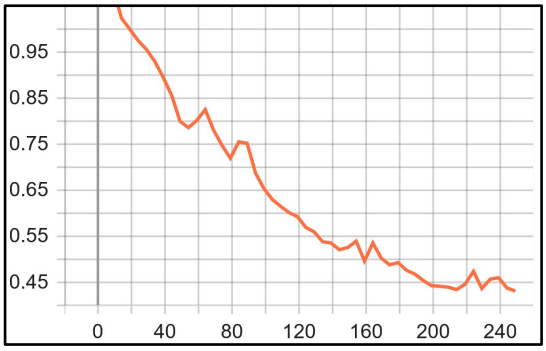

Figure 10: Validation loss of DETR for classification on KAIST

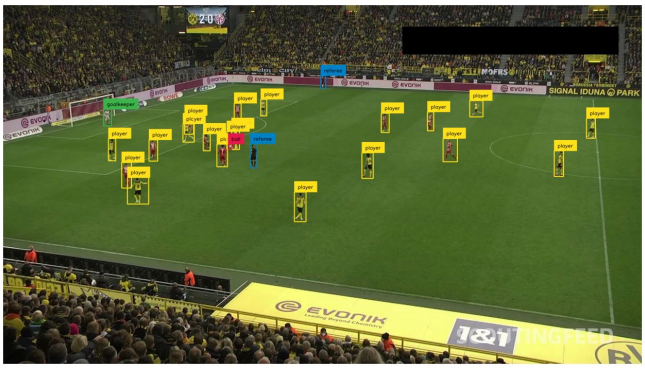

Figure 11: Image_compared_to_detection

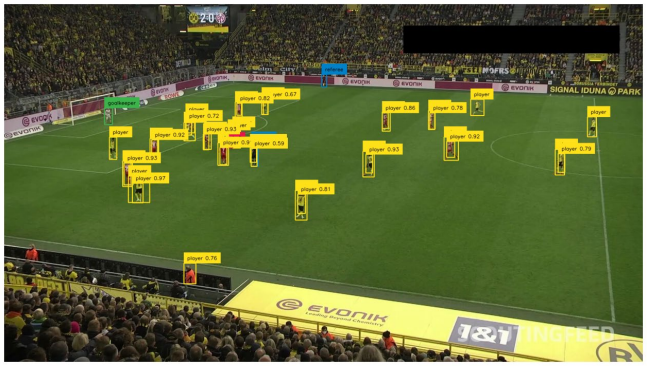

Figure 12: Image_comp_actual_detected

A few challenges were faced during implementation and execution of our updated model architecture. The utilization of the KAIST dataset poses significant challenges due to its extensive size, demanding substantial computational time and resources for processing. Notably, the dataset employs XML format annotations, differing from the COCO format used in DETR. While the original DETR model features a single encoder and decoder block, our adaptation for KAIST requires a comprehensive modification of the transformer architecture, particularly in the encoder. This adjustment involves the incorporation of two parallel modalities, requiring the development of custom encoders different from the standard DETR ones. Consequently, aligning the decoders becomes a complex task, making it challenging to determine the optimal number of layers for processing while maintaining attention between the encoders. Working with tensors and ensuring consistent input-output dimensions across layers further complicates the process. Additionally, finding a perfectly aligned dataset proves difficult, and even when discovered, inadequate annotations require bounding before effective training can take place. Lastly, attempting training on a small dataset proves futile, as it leads to degraded model performance and increases the likelihood of inaccuracies, rendering the results of the model unreliable.

V. CONCLUSION

The purpose of our research was to develop a transformer model that allows for the seamless integration of multiple modalities in order to improve the prediction of object detection models. Through our experimentation, we were able to leverage our modified DETR transformer model to capture and process image data from two localizations of the electromagnetic and light spectrum: RGB and IR. We fused these modalities features into the updated model which successfully performed object detection.

Through our results and findings, we were able to prove that our updated architecture performed slightly better

than the standard DETR database in performing object detection. Our new model can help improve multiple sectors from healthcare, autonomous driving, to even military applications to support newer and more advanced object detection and processing techniques.

Instead of concatenating the token's features, we can utilize different encoders to focus on individual modality. We can build two parallel ResNet50, with two parallel encoders, with a selection unit and a decoder. Although this would increase the model size with more parameters, we can build a powerful architecture to focus on individual modalities depending on our requirement by using selection mux to select that modality. Therefore, we can reap the benefits of enhanced features due to cross-attention, to adapt to missing modality, as well as perform optimal modality integration.

In our future work, we aim to expand the current framework from two modalities to three modalities, enhancing our capability to capture and integrate information from multiple sources comprehensively. Additionally, we propose a new architecture that incorporates two encoders. This development will enable us to process the data features of different modalities with greater precision. By introducing a second encoder, we can provide dedicated feature extraction and encoding for each modality, thereby improving the overall expressiveness and accuracy of the model. Such advancements will lay a solid foundation for addressing more complex issues such as cross-modal understanding and generation.

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)

APPENDIX

In our conclusion, we propose a new architecture with two encoders, as illustrated in the following flowchart:

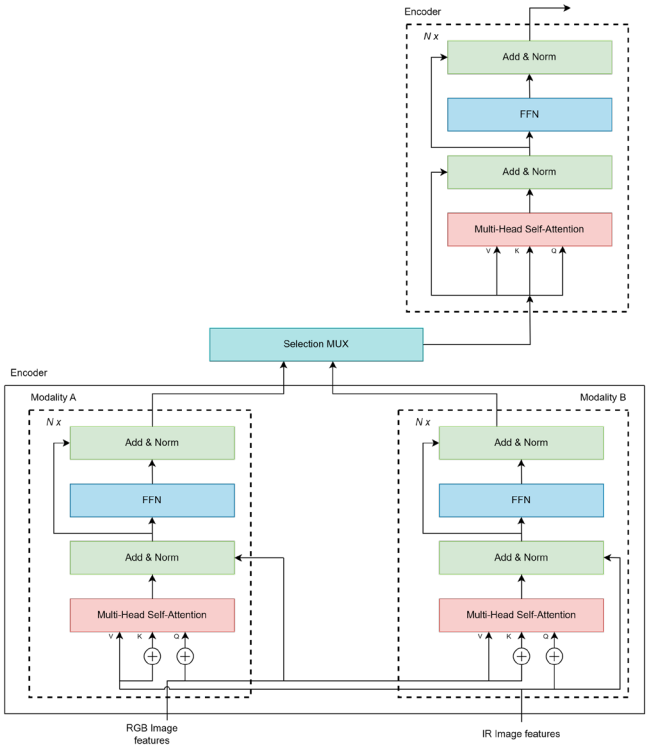

Figure 13: Newly Proposed DETR Architecture