TDYSN: A YOLO-Based Top Down Saliency Prediction Network

Can Mızraklı* and Tolga Çapın†
Department of Computer Engineering, TED University, Ankara, Turkey
*can.mizrakli@tedu.edu.tr, †tolga.capin@tedu.edu.tr

***Abstract*—Visual saliency seeks to identify the most striking parts in an image. Top-down saliency prediction refines this by including higher-level task definitions. YOLO has set the standard in computer vision with its processing speed, detection accuracy and computational efficiency. Our contribution in this paper involves proposing an architecture with a pre-trained YOLO backbone for feature extraction. Coupled with a feature projection module for channel-compression of high-level features, a Sentence-BERT-based task encoder derives the semantic embeddings from the text-based task description, which are then merged with the visual features using a transformer fusion module. It is then decoded into the resulting saliency maps with the task-relevant areas highlighted.**

***Index Terms*—Visual Saliency, Top-Down Saliency, YOLO, Feature Projection Module, Sentence-BERT, Transformer Fusion**

I. Introduction

Visual saliency seeks to identify the most attention-grabbing regions in an image. Traditional bottom-up methods rely on low-level cues like contrast and edges to produce general-purpose saliency maps, but they neglect high-level goals. Task-driven (top-down) saliency prediction addresses this limitation by incorporating explicit task definitions such as "count people" or "find objects", to guide attention toward context-specific regions.

In this paper, we introduce **TDYSN**, a modular deep network that fuses visual and semantic cues in three stages: (1) a pre-trained YOLO [1] backbone for efficient, multiscale feature extraction; (2) a 1×1 convolutional Feature Projection Module (FPM) and a Sentence-BERT task encoder to compress visual maps and encode textual prompts into embeddings; and (3) a lightweight transformer fusion block that jointly processes the visual–semantic tokens, followed by a decoder that reconstructs the fused representation into a final saliency map highlighting task-relevant areas.

The rest of the paper is organized as follows: Section II reviews related work on top-down saliency and vision–language fusion. Section III details the TDYSN architecture and training protocol. Section IV presents quantitative and qualitative evaluations. Finally, Section VI concludes and outlines future directions.

A. Contributions

This paper makes three key contributions:

- We introduce **TDYSN**, a top-down saliency model that fuses YOLO-derived visual features with Sentence-BERT task embeddings via a lightweight transformer block.
- We train and evaluate TDYSN on a 1,968-image, four-task eye-tracking dataset, achieving state-of-the-art NSS and AUC-Borji scores while preserving end-to-end efficiency.
- We validate our approach through both quantitative metrics (CC, KL, SIM, NSS, AUC) and qualitative examples, and show how TDYSN can be adapted for practical, attention-aware applications.

II. Related Work

For a long time, visual saliency has been a core issue in computer vision. Until recently, bottom-up approaches were the primary methods to deal with visual saliency, but later studies started to work on top-down saliency approaches. In this section, more recent work that has driven the field and applied top-down approaches to predict visual saliency are reviewed.

A. A Study of Visual Saliency for Free-Viewing and Task-Oriented Condition

Dilara Albayrak's *A Study of Visual Saliency for Free-Viewing and Task-Oriented Condition* analyzes the concept of saliency when varying viewing conditions are present. The thesis uses Generative Adversarial Networks (GANs) to improve the prediction of saliency maps by learning hierarchical representations that cover semantic and context-based information [2].

The data used in our project comes from the data gathered by Albayrak's work. It forms the base of our work by providing ground truth saliency maps based on four different high-level task definitions.

B. Visual Saliency Transformer

The Visual Saliency Transformer (VST) by Liu et al. uses a transformer-based backbone for saliency detection to overcome the limitations of conventional CNNs. VST tokenizes images and uses multihead self-attention for global interdependencies. It adopts a reverse T2T token upsampling to form a high-resolution output and a joint learning-based multitask decoder to predict saliency maps with boundary maps, utilizing task tokens and patch-task attention. Experiments demonstrate

the superiority of performance compared to other state-of-the-art CNN-based architectures [3].

This paper inspired our model to integrate transformers, but in a slightly different way, by creating a transformer fusion module to merge visual features with textual task embeddings while leveraging YOLO's feature-extracting abilities.

C. Top-Down Saliency Detection Driven by Visual Classification

In *Top-Down Saliency Detection Driven by Visual Classification*, the idea of generating saliency maps through classification is introduced by the authors. They establish high-level information for tasks that can effectively guide the identification of regions most relevant for classification [4].

D. Visualising Image Classification Models and Saliency Maps

Simonyan et al. introduced a gradient-based method for visualizing image classification network saliency. Their work consists of calculating the derivative of the class score with respect to input pixels, which indicates which regions inside the image have the largest impact on a CNN's output. While this technique is centered on general interpretability as opposed to explicit task-oriented cues, it serves as the foundation for gradient-based methods. [5].

E. Literature Discussion

The reviewed works highlight the effectiveness of incorporating high-level, task-specific definitions into saliency prediction. For example, Dilara Albayrak's work illustrates that incorporating task-oriented viewing conditions can enhance saliency prediction [2]. Similarly, Murabito et al. [4] shows how utilizing classification can guide the detection of regions most relevant for a defined task. Additionally, the Visual Saliency Transformer [3] demonstrates that transformer architectures can capture the global dependencies necessary to integrate high-level tasks into saliency maps.

While existing methods either rely on generic saliency cues or pure transformer pipelines, our TDYSN model fuses a YOLO-based detector, a Sentence-BERT task encoder, and a transformer fusion module to generate task-aligned saliency maps. In the next section, we detail its architecture and training.

III. Proposed Method

Our goal is to create a top-down saliency prediction model that identifies visually striking regions to the human eye based on a task. By integrating a pre-trained YOLO-based architecture for feature extraction with a Sentence-BERT-based task encoder, our methodology bridges the gap between visual features and semantic task definitions. The YOLO backbone performs well at extracting visual cues, and the Sentence-BERT encoder transforms high-level text-based task descriptions into semantic embeddings. Then the transformer-based fusion module processes these two components together to create a connection between extracted features based on tasks. Consequently, the model benefits from the strengths of both object detection and contextual understanding, resulting in more accurate task-specific saliency maps.

A. Dataset & Preprocessing

The dataset used here was originally collected by Albayrak [2]. It comprises one folder of original stimuli images and, for each of four task conditions, two subfolders: `fdm` (grayscale eye-tracker saliency maps) and `heatmap` (RGB overlays on the stimuli).

In our experiments, we use only the raw grayscale maps from the `fdm` subfolders as ground truth, yielding a total of 1,968 image–saliency pairs across the four tasks. We split these into:

- 70% train (1,377 pairs)
- 15% validation (295 pairs)
- 15% test (296 pairs)

All splits are generated with a fixed random seed (42) to ensure reproducibility.

Preprocessing and augmentations are applied identically to each image–map pair:

- **Resize:** Stimuli to 384×384 px. Ground-truth FDMs are recorded at 48×48; we keep them at that size and later down-sample predictions to 48×48 before computing the loss.
- **Normalization:** scale image pixel values to $[0, 1]$ and convert to tensors.
- **Paired augmentations:** random horizontal flip (probability 0.5) and random rotation within $\pm 10°$, applied the same way to the image and its corresponding map to preserve spatial correspondence.

B. Model Architecture

The architecture consists of a YOLO-based backbone for visual feature extraction, an FPM (1×1 conv projection), a task encoder to generate semantic embeddings, a transformer fusion module that combines visual and semantic features, and a final saliency decoder that produces the output saliency maps (see Figure 1).

1) YOLO Backbone: The backbone contains a pre-trained YOLO model [1] to extract high-level features from the images. The backbone is responsible for capturing edges, textures, and object parts that are fundamental for scene understanding. We focused on using the earlier layers of YOLO, which are optimized for object detection, helping our model benefit from its pre-trained general feature extraction capabilities.

2) Feature Projection Module (FPM): The FPM applies a single 1×1 convolution to the high-dimensional feature maps outputted by the YOLO backbone, reducing their channel dimensionality from 512 to 128, while preserving spatial resolution. By compressing the feature representation in this way, the FPM distills the most salient visual information into a more compact form, which both lowers computational overhead and facilitates seamless fusion with the task embeddings in

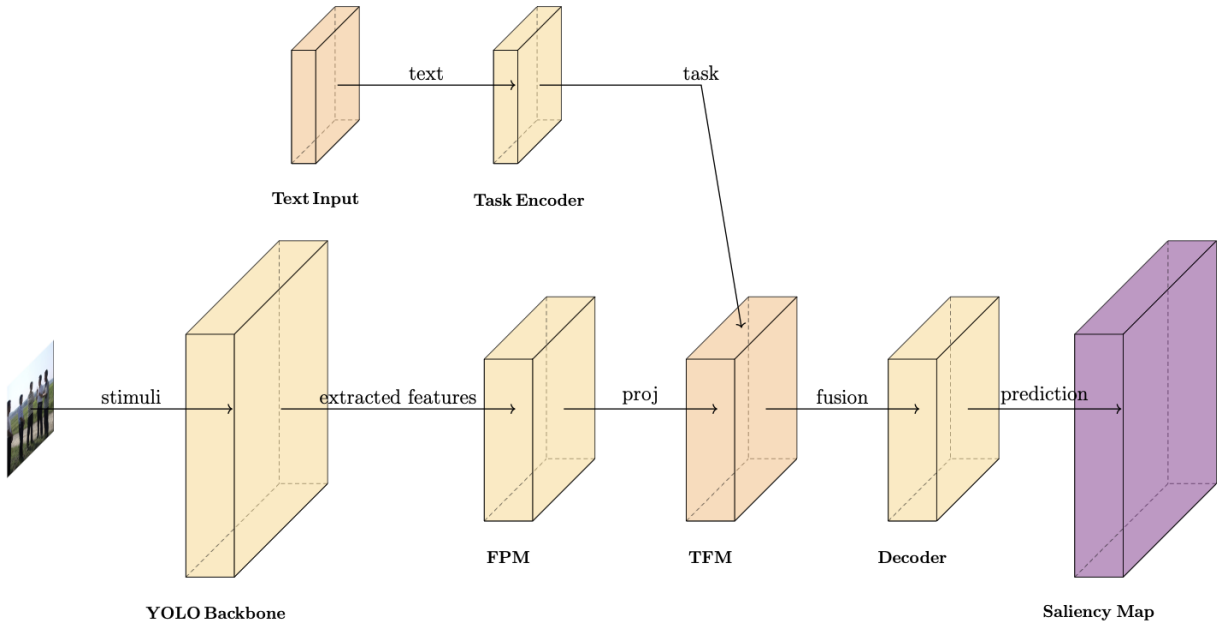

Fig. 1: Overview of the model architecture.

the transformer module, ultimately improving efficiency and effectiveness of the saliency decoder.

3) Task Encoder: To utilize task-specific information, the task encoder represents task descriptions in dense semantic embeddings. Using Sentence-BERT, the model maps text-based descriptions to vector representations. This is further fine-tuned through a linear transformation to condense the information into a compact representation that can be easily blended with visual features. This bridges the language-vision gap so that the model can recognize and focus on regions in the picture that are meaningful to the given task.

4) Transformer Fusion Module (TFM): TFM fuses the visual and semantic modalities produced before. First, feature maps outputted from FPM are transformed into a sequence of tokens, flattening the spatial dimensions. The encoded task embeddings, which represent the task definitions, are added at the end of this sequence as additional tokens. The transformer encoder processes these combined tokens, which allows it to learn complex dependencies between visual features and task definitions. The end product is a representation that emphasizes the areas of the image that are critical for that specific task. This fused sequence is then reshaped back into spatial format for further processing.

5) Saliency Decoder: The saliency decoder translates the fused feature map into a single-channel saliency map. It uses a series of convolutional layers that reconstruct and refine the spatial information. A sigmoid activation function at the output normalizes the resulting map between 0 and 1, representing the probability of each pixel being salient. This transforms the fused representation into an actual output that can be directly interpreted as a task-driven saliency map prediction.

TABLE I: Implementation Details

Component	**Key Settings**
Backbone	YOLOv5-su (Ultralytics) first 10 layers (SPPF output, 512-ch)
Feature Projection Module (FPM)	1×1 conv: 512→128 channels output $[B, 128, H, W]$
Task Encoder	SentenceTransformer (all-MiniLM-L6-v2): 384→64-dim
Transformer Fusion	1-layer Transformer Encoder $d_{\text{model}} = 128,\ n_{\text{head}} = 4$ + 1 task token prepended
Saliency Decoder	Conv $3 \times 3 \rightarrow 64 \rightarrow$ Deconv×2→32 $\rightarrow$ Deconv×2→1 Sigmoid $\rightarrow [B, 1, H, W]$
Reproducibility	Random seed 42 for dataset splitting

C. Training Configuration

We train our TDYSN model for 40 epochs. The optimizer is Adam with a constant learning rate of 1×10^{-4} and no additional weight decay. We use a batch size of 8 and reset gradients at each step.

The loss is a combined saliency objective:

$$\mathcal{L}(S, \hat{S}) = \alpha\, \mathrm{KL}(S \,\|\, \hat{S}) \;+\; \beta\big(1 - \mathrm{CC}(S, \hat{S})\big),$$

with weighting factors $\alpha = \beta = 1.0$. Here, KL stands for the *Kullback–Leibler divergence* [6] [7], which measures the discrepancy between the predicted saliency distribution $\hat{S}$ and the ground-truth distribution S, and CC stands for the *Pearson correlation coefficient* [7] [8] , interpreted as the cosine of the angle between the predicted and ground-truth saliency maps.

At each forward pass, the model outputs a 96×96 saliency map, which we bilinearly downsample to 48×48 to match the ground-truth eye-tracker maps before computing the loss. We

log both batch-level and epoch-average loss values to monitor convergence (Figure 2).

IV. Results

A. Quantitative Evaluation

To evaluate the performance of our TDYSN model, we computed five widely adopted saliency metrics (Figure 3 & Table II) across the validation set: Pearson's Correlation Coefficient (CC) [7] [8], Kullback-Leibler Divergence (KLDiv) [6] [7], Similarity (SIM) [7], Normalized Scanpath Saliency (NSS) [7], and AUC-Borji [9]. These metrics reflect different aspects of saliency quality: CC measures spatial correlation, KLDiv measures distribution divergence, SIM captures global map similarity, NSS evaluates alignment with fixation points, and AUC-Borji measures discriminative power between salient and non-salient regions (Figure 3).

TABLE II: Final performance metrics on the validation set

Metric	Value
CC	0.6433
KLDiv	0.9239
Similarity	0.5112
NSS	3.5290
AUC-Borji	0.9489

Our model achieves a NSS of 3.53 and AUC-Borji of 0.95, both of which are highly competitive results in the literature. This comparison to MIT/Tübingen Saliency Benchmark [10] is later discussed in the Comparison to Literature section (V-B).

We follow the saliency evaluation methodology outlined by Kümmerer et al. [7], which emphasizes separating models, maps, and metric computation. Although our dataset differs from MIT300 and SALICON, the results demonstrate strong generalization and alignment with benchmark expectations.

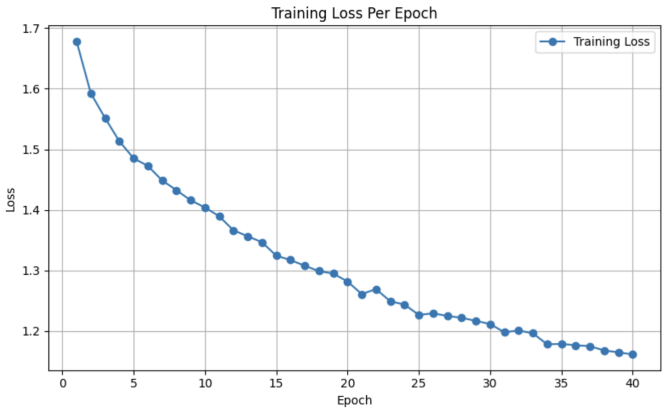

Fig. 2: Training loss per epoch for 40 epochs.

B. Training Convergence

Figure 2 illustrates the training loss across 40 epochs. The model shows a consistently decreasing trend in loss, dropping from an initial value above 1.65 to approximately 1.18 by the final epoch. This steady decline suggests effective learning throughout training. Importantly, the curve exhibits no significant oscillations or spikes, indicating there is not any notable instability or overfitting. The rate of improvement slows down around epoch 30, suggesting convergence as the model begins to saturate in performance.

Overall, the smooth downward trajectory of the loss curve confirms that the training process is both stable and efficient, validating our architectural and optimization choices.

C. Metric Trends Over Time

Figure 3 shows the trends of five validation metrics: NSS, AUC-Borji, CC, SIM, and KL divergence, over the 40 training epochs.

The trend that stands out the most is the steady and consistent increase in NSS, which starts around 2.65 and rises to 3.75 by epoch 40. This indicates that the model becomes increasingly aligned with human fixation patterns as training progresses. Similarly, AUC-Borji shows a gradual upward trend, improving from approximately 0.91 to 0.95, suggesting enhanced capability to rank salient regions correctly.

The CC and SIM metrics, also show incremental increase, indicating better global structure matching. Meanwhile, KL divergence steadily decreases from about 1.15 to below 0.95, reflecting reduced distributional mismatch between predicted and ground-truth saliency maps.

Overall, the trends across all metrics confirm that the model not only converges, but also generalizes better with longer training, with NSS and AUC-Borji showing the most significant improvements.

D. Qualitative Visualization

To better understand the task-driven behavior of our TDYSN model, we present saliency predictions on four different sample images under distinct task conditions. Each row in Figure 4 corresponds to a different task prompt, showing the ground-truth saliency map (FDM), and the predicted saliency map.

As seen in the results, the predicted saliency maps shift focus based on the given task prompt, demonstrating the model's ability to incorporate task semantics into spatial attention. The inclusion of the original stimuli images with ground truths as heatmaps on them helps illustrate how these attention shifts are contextually grounded in the visual scene, further validating the task-driven behavior of our model.

V. Discussion

A. Interpretation of Results

The results presented in Section 4 confirm that our TDYSN model is capable of learning highly accurate, task-dependent saliency representations. The model achieved a NSS score of 3.53 and an AUC-Borji score of 0.9489, both of which indicate strong alignment with human fixation patterns. These metrics increased steadily across the 40 training epochs, suggesting stable convergence and continuous learning of task-relevant visual features. The upward trajectory in NSS is particularly notable, as it directly measures how well predicted saliency values align with human fixations at discrete locations. The

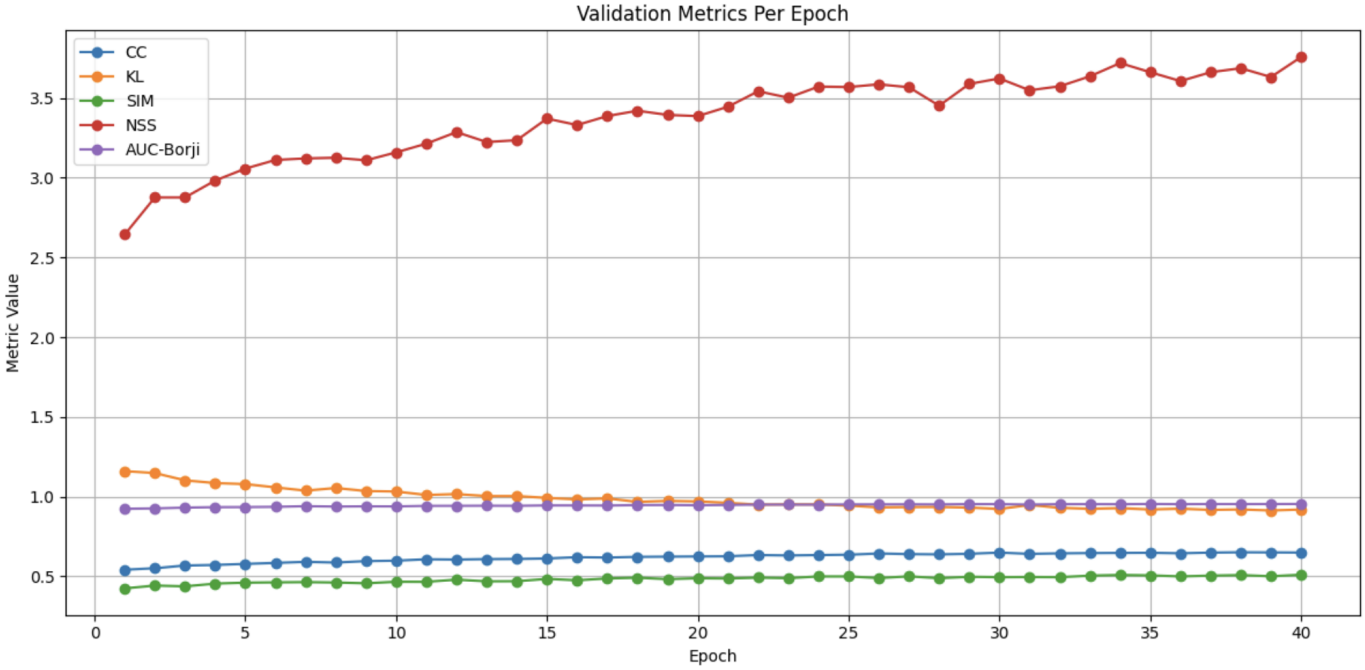

Fig. 3: Validation metrics (CC, KL, SIM, NSS, AUC–Borji) over 40 epochs.

consistent improvement in AUC-Borji further reinforces the model's ability to discriminate between salient and non-salient regions.

The CC and SIM scores, which are 0.6433 and 0.5112 respectively, reflect improved spatial structure preservation and histogram overlap, which are important for ensuring that predicted saliency maps capture both local focus and global attention cues. Meanwhile, KL divergence dropped steadily to 0.9239, reflecting reduced distributional mismatch between predictions and ground truth over time. Together, these trends indicate that the model not only converges smoothly but also generalizes its attention mapping across tasks and image types.

B. Comparison to Literature

TABLE III: Performance comparison between TDYSN, EML-Net, and the Gold Standard from MIT/Tübingen Saliency Benchmark [10].

Model	**NSS**	**AUC**	**CC**	**KLDiv**	**SIM**
TDYSN (Ours)	3.53	0.9489	0.6433	0.9239	0.5112
EML-Net	2.05	0.8660	0.8860	0.5200	0.7800
Gold Standard	3.14	0.9341	0.9828	0.0602	0.8992

Our performance compares favorably with existing models in the saliency literature. According to the MIT/Tübingen Saliency Benchmark [10], the Gold Standard achieves an NSS of 3.14 and an AUC-Borji of 0.9341 on the MIT300 dataset. These values serve as reference points for high-performing general-purpose saliency models. We also compare against EML-Net [11], a strong CNN-based model evaluated under free-viewing conditions, which reports an NSS of 2.05 and an AUC of 0.866.

Although our TDYSN model is trained on a task-driven dataset and not directly benchmarked on MIT300, it surpasses both baselines in NSS (3.53) and AUC-Borji (0.9489), suggesting superior alignment with human fixations even task definitions. Higher CC/SIM in Gold Std. reflect smoother maps; task-conditioned focus decreases these metrics but improves NSS/AUC. While TDYSN underperforms in CC and SIM—metrics that favor smoother and more globally correlated predictions, this is expected due to the increased complexity of modeling attention in goal-directed tasks where saliency is conditioned on semantic intent rather than general visual prominence.

C. Limitations

Despite promising results, several limitations remain. First, our dataset comprises only 1,968 image–task pairs across four fixed task categories. While sufficient for demonstrating feasibility, this limited scale and semantic diversity may constrain the model's ability to generalize to unseen task types or domains.

Second, we did not perform cross-dataset validation on datasets like SALICON or MIT300 since we do not have the gaze map data collected on those benchmark datasets, which would be necessary to rigorously assess generalization beyond our training distribution. Third, our model assumes the availability of structured, sentence-level task descriptions. In real-world applications, such task inputs may be noisy, ambiguous, or unavailable altogether, which could reduce prediction accuracy.

Finally, although our model outperforms established baselines in metric terms, we did not perform a detailed ablation

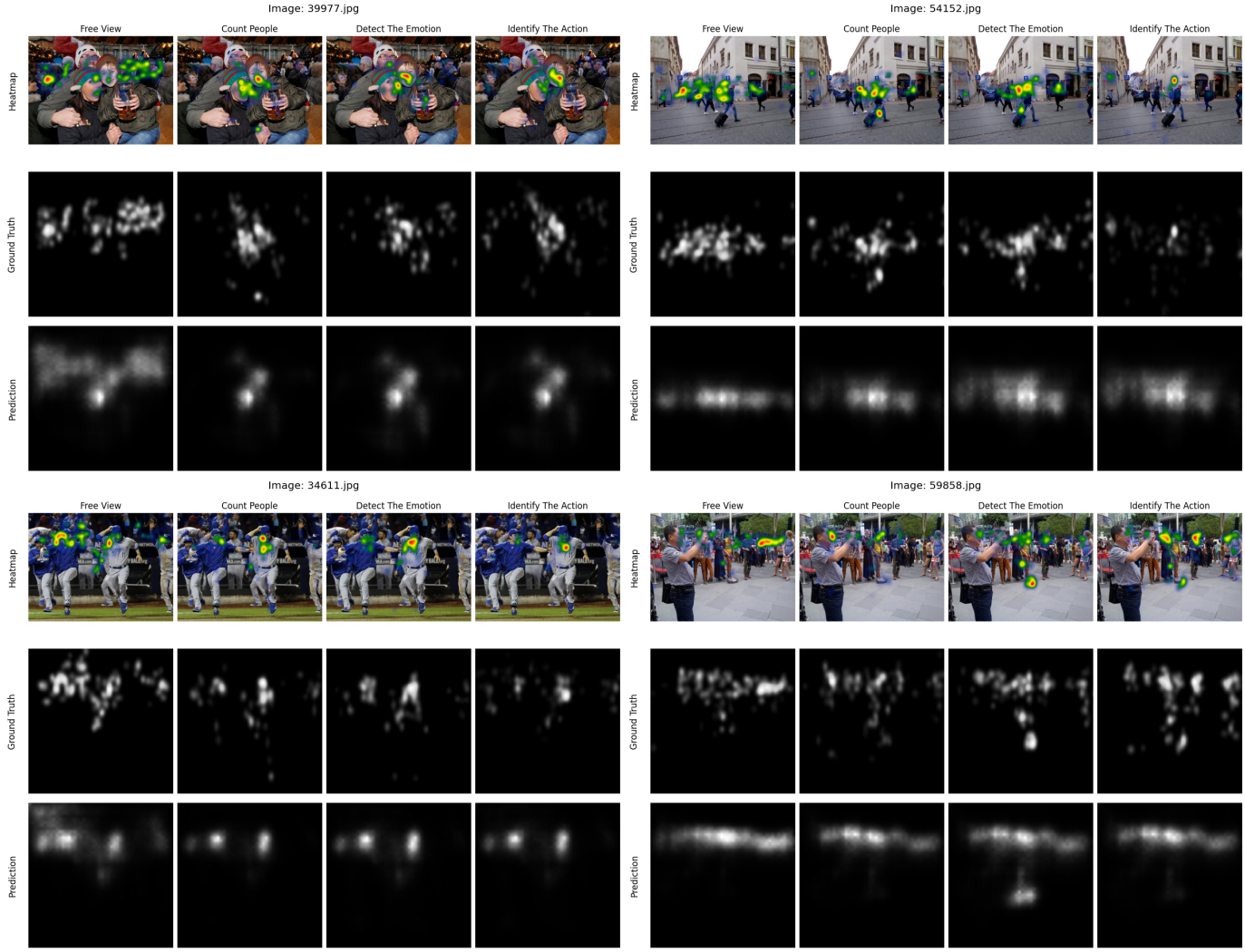

Fig. 4: Example saliency predictions for 4 random images for different task conditions. Each row: prediction and ground truth saliency maps.

study to isolate the contribution of each architectural component (e.g., task encoder, transformer fusion). This leaves open the question of which elements are most critical to performance.

D. Future Work

For future work, we aim to extend the scope of our model in two ways: first, by expanding our existing dataset from 1,968 images through additional data collection following the same protocol used by Albayrak [2]. Secondly, by rearchitecting key components of the model, such as experimenting with multi-layer transformer fusion or task-specific decoding blocks to further enhance its ability to adapt to a broader range of task semantics and visual scenarios.

Second, we will perform an ablation study to analyze the relative contribution of each component (YOLO backbone, transformer fusion, task encoder). This will help identify minimal yet effective configurations for real-time deployment.

These efforts aim to improve the model's generalization, scalability, and interpretability, ultimately enabling more robust task-driven saliency prediction across diverse real-world conditions.

VI. Conclusion

We presented TDYSN, a lightweight yet effective top-down saliency prediction network that couples a pre-trained YOLO backbone with a 1×1 Feature Projection Module, a Sentence-BERT task encoder, and a transformer-based fusion block. On Albayrak's task-oriented eye-tracking dataset [2] TDYSN attains an NSS of 3.53 and AUC-Borji of 0.9489, exceeding both the Gold Standard [10] and EML-Net [11] while maintaining a compact, end-to-end trainable design. Consistently rising metric curves and qualitative visualisations confirm that the model not only converges with stability but also shifts attention in line with the provided task prompts, which proves that integrating high-level semantics with multi-scale visual cues is crucial for goal-directed saliency.

Despite these encouraging results, TDYSN is currently bounded by the scale and semantic breadth of the available data and has yet to be validated across heterogeneous saliency benchmarks. Future work will therefore focus on expanding and diversifying the dataset, performing cross-dataset studies, and modifying the architecture for enhanced performance. We believe these directions will extend TDYSN's applicability to real-world settings ranging from assistive vision and human–robot interaction to attention-aware image understanding, and will stimulate further research at the intersection of visual attention, language grounding, and efficient detection backbones.

REFERENCES

[1] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 779–788.

[2] D. Albayrak, "A study of visual saliency for free-viewing and task-oriented condition," Master's thesis, TED University, 2020.

[3] N. Liu, N. Zhang, K. Wan, L. Shao, and J. Han, "Visual saliency transformer," in *Proc. IEEE/CVF Int. Conf. on Computer Vision (ICCV)*. Montreal, QC, Canada: IEEE, 2021, pp. 4722–4732.

[4] F. Murabito, C. Spampinato, S. Palazzo, D. Giordano, K. Pogorelov, and M. Riegler, "Top-down saliency detection driven by visual classification," *Computer Vision and Image Understanding*, vol. 172, pp. 67–76, 2018.

[5] K. Simonyan, A. Vedaldi, and A. Zisserman, "Deep inside convolutional networks: Visualising image classification models and saliency maps," *arXiv*, Dec. 2014.

[6] J. Liang and Y. Zhang, "Top down saliency detection via kullback–leibler divergence for object recognition," in *Proceedings of the 4th International Symposium on Bioelectronics and Bioinformatics (ISBB)*. Beijing, China: IEEE, 2015, pp. 200–203.

[7] M. Kümmerer, T. S. Wallis, and M. Bethge, "Saliency benchmarking made easy: Separating models, maps and metrics," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018.

[8] Z. Gniazdowski, "Geometric interpretation of a correlation," *Zeszyty Naukowe Warszawskiej Wyższej Szkoły Informatyki*, vol. 9, no. 7, pp. 27–35, 2013.

[9] A. Borji, D. N. Sihite, and L. Itti, "Quantitative analysis of human-model agreement in visual saliency modeling: A comparative study," *IEEE Transactions on Image Processing*, vol. 22, no. 1, pp. 55–69, Jan 2013.

[10] M. Kümmerer, T. Wallis, and M. Bethge, "Mit/tübingen saliency benchmark," 2024, https://saliency.tuebingen.ai/results.html.

[11] X. Jia and N. D. B. Bruce, "Eml-net: An expandable multi-layer network for saliency prediction," in *Proc. IEEE/CVF Conf. on Computer Vision and Pattern Recognition Workshops (CVPRW)*. Seattle, WA, USA: IEEE, 2020, pp. 4429–4438.