# Domain-Specialized Object Detection via Model-Level Mixtures of Experts

Svetlana Pavlitska[1,2], Malte Stüven[2], Beyza Keskin[2], and J. Marius Zöllner[1,2]
[1]*FZI Research Center for Information Technology* [2]*Karlsruhe Institute of Technology (KIT)*
pavlitska@fzi.de

***Abstract*—Mixture-of-Experts (MoE) models provide a structured approach to combining specialized neural networks and offer greater interpretability than conventional ensembles. While MoEs have been successfully applied to image classification and semantic segmentation, their use in object detection remains limited due to challenges in merging dense and structured predictions. In this work, we investigate model-level mixtures of object detectors and analyze their suitability for improving performance and interpretability in object detection. We propose an MoE architecture that combines YOLO-based detectors trained on semantically disjoint data subsets, with a learned gating network that dynamically weights expert contributions. We study different strategies for fusing detection outputs and for training the gating mechanism, including balancing losses to prevent expert collapse. Experiments on the BDD100K dataset demonstrate that the proposed MoE consistently outperforms standard ensemble approaches and provides insights into expert specialization across domains, highlighting model-level MoEs as a viable alternative to traditional ensembling for object detection. Our code is available at https://github.com/KASTEL-MobilityLab/mixtures-of-experts/**

***Index Terms*—mixture of experts, object detection**

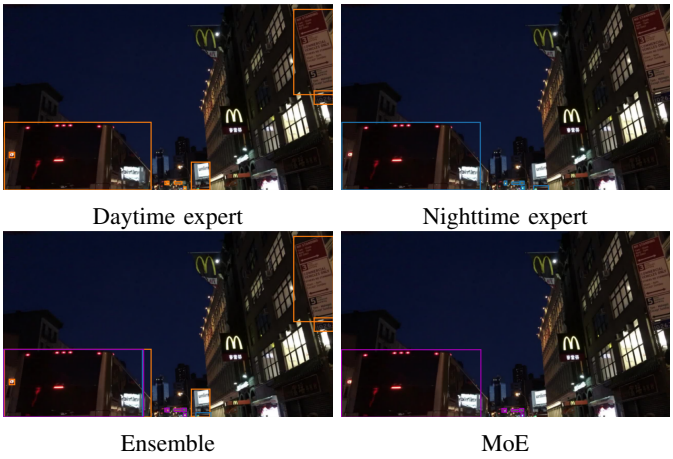



Fig. 1: Predictions of different models on an exemplary image from the BDD100K dataset.

## I. Introduction

Object detection systems are increasingly deployed in safety-critical and real-world settings where both predictive accuracy and interpretability are essential. Ensemble methods are used to improve detection performance. However, in standard object detector ensembles where predictions from all models are simply concatenated and a suppression-based postprocessing step is applied, interpretability is inherently limited. The final decision emerges from heuristic postprocessing rather than from an explicit model-level mechanism that explains how individual detectors contribute. Once predictions are merged, it is difficult to attribute a retained detection to a specific expert, to understand why one expert dominated over another, or to analyze how expert behavior varies across input conditions. Non Maximum Suppression (NMS) further obscures attribution by discarding overlapping predictions without preserving information about suppressed alternatives or their originating models. Ensembles thus primarily improve performance but provide little insight into model agreement, disagreement, or domain-specific specialization.

Mixture-of-experts (MoE) [1] models offer a principled alternative by combining multiple specialized networks through an explicit routing mechanism that enables both performance gains and interpretability through expert selection.

Recent work has made substantial progress on layer-level MoE architectures for transformer models [2]–[5], where expert modules are selectively activated within individual network layers to efficiently scale model capacity, and these approaches have been particularly successful in large language models. In contrast, studying model-level MoEs remains important because it enables specialization across complete models with distinct inductive biases or training domains and provides clearer interpretability and modularity that are difficult to achieve when expert behavior is distributed across internal layers.

Model-level MoE architectures have been extensively studied in image classification and semantic segmentation, where predictions are dense and structurally aligned [6]–[9]. In contrast, their application to object detection has received limited attention due to challenges in combining structured and variable-length predictions, such as bounding boxes (BBoxes) and confidence scores. These challenges make it nontrivial to apply expert weighting prior to detection postprocessing while preserving the benefits of specialization across experts.

In this work, we investigate model-level MoEs for object detection and analyze their effectiveness as an alternative to conventional ensembling. We propose an MoE architecture that combines YOLO [10]-based detectors trained on semantically disjoint data subsets, using a learned gating network that dynamically weights expert contributions at inference time. To address detection-fusion challenges, expert weighting is applied prior to postprocessing, and several strategies for merging BBox

predictions are evaluated.

Experiments are conducted on the BDD100K dataset [11] using a split based on daytime and nighttime conditions to induce expert specialization. Results show that the proposed MoE consistently outperforms standard ensemble methods.

Following the distinction between interpretability and explainability [12], [13], we focus on inherent interpretability, where the decision process is directly understandable from the model structure, and the proposed MoE provides it by explicitly exposing expert contributions, in contrast to ensembles that rely on opaque post hoc fusion. In addition to quantitative gains, our analysis of expert routing and prediction disagreement provides insight into domain-dependent expert behavior, demonstrating that model-level MoEs can improve both performance and interpretability in object detection.

## II. Related Work

### A. Ensembles of CNNs for Object Detection

Multiple strategies have been proposed for fusing predictions from several detectors. A common baseline aggregates BBox predictions from all models and applies class-level suppression to remove redundant detections. While this approach is efficient and architecture-agnostic, it may discard useful information when overlapping boxes are slightly misaligned.

More advanced fusion methods explicitly combine spatially overlapping detections into a single prediction. All such approaches treat predictions from different detectors as if they originated from a single model and group BBoxes whose overlap exceeds a fixed threshold. Weighted Boxes Fusion (WBF) computes confidence weighted averages of BBox coordinates and adjusts the final score based on the number of contributing models [14]. Non Maximum Weighted (NMW) fusion applies weights derived from both confidence and spatial overlap with the most confident prediction in each group [15]. Soft NMS decays confidence scores in proportion to spatial overlap with higher scoring detections [16]. These methods allow consistent predictions across heterogeneous architectures to reinforce one another and often improve ensemble accuracy.

Following the approach of Xu et al. [17], ensemble predictions can be formed by aggregating final BBoxes from all experts and applying a suppression-based selection mechanism to produce a single prediction per object.

Casado et al. distinguish ensemble methods that operate within the detection algorithm from those that combine model outputs [18]. Algorithm-level approaches include feature fusion prior to region proposal generation [19], [20], classification-stage ensembling [21], [22], and methods that integrate multiple stages of the detection pipeline [23]. Output-level ensembles combine predictions from different detectors, such as Fast R-CNN and Faster R-CNN [24], YOLO-based models [10], or RetinaNet and Mask R-CNN [14].

Bahhar et al. present a wildfire and smoke detection framework that combines a voting-based classifier with a two-stage YOLO detector [25]. Images are processed in parallel by both components, and localization is performed only when the classifier predicts the presence of fire or smoke using YOLOv5 [26]. The method achieves strong precision and recall with fast inference suitable for edge deployment, but depends on high-quality imagery and may produce false positives in challenging scenarios [27].

Huayhongthong et al. propose an ensemble framework for incremental object detection to reduce training cost [28]. The system combines a pre-trained detector with a transferred model trained on new object classes using YOLO as the base architecture. An ensemble decision module selects the final prediction by comparing class probabilities from both models, enabling the addition of new classes without retraining the original detector [28].

### B. Mixture of Experts

Mixture of experts originates from the adaptive mixture of local experts framework by Jacobs et al. [1], where a learned gating function combines the outputs of multiple experts, each of which is a complete model that can specialize on different regions of the input space. In computer vision image classification, this model-level mixture idea has been explored as a learnable alternative to standard ensembles, and it has proven especially relevant for fine-grained recognition, where different experts can focus on complementary visual cues. A representative example is the mixture of granularity-specific experts' approach for fine-grained categorization, which explicitly trains experts at different granularity levels and fuses their predictions via learned weighting [29]. Very recent work continues this direction for fine-grained visual classification with heterogeneous experts and collaborative gating mechanisms [30].

For dense prediction, our previous work [7] applies model-level mixtures to semantic segmentation by training separate segmentation networks as experts and learning a gate to fuse their outputs, emphasizing interpretability and insight into expert behavior across scenes. A closely related and influential vision line is the work by Valada et al. [6], which uses a mixture of deep experts for robust semantic segmentation and also exemplifies a common application setting for model-level mixtures in vision: multisensor fusion, where experts operate on different modalities and the mixture learns how to combine them depending on the environment.

This multisensor pattern also appears in object detection, for example, in adaptive multimodal fusion approaches that combine modality-specific expert detectors, such as RGB, depth, and optical flow, to improve robustness under changing conditions [9]. MoE dynamically adapts to changing environment settings, assigning higher weights to the most confident expert.

For standard single-modality object detection, MoCaE highlights that combining multiple complete detectors can be limited by expert miscalibration and proposes calibrated expert fusion to better exploit complementary strengths across detection benchmarks [31]. While MoCaE focuses on calibrating heterogeneous detectors to prevent confidence-dominated fusion, our work studies model-level mixtures with learned gating, highlighting domain-aware routing and interpretability as key challenges and opportunities for MoEs in object detection.

## III. Approach

Building upon our previous works [7], [32], we propose an MoE architecture comprising several pre-trained CNNs for object detection, each fine-tuned on a subdomain of the input distribution. Their predictions are then combined dynamically during inference.

### A. MoE Architecture

Formally, an MoE model consists of $n$ individual object detection experts $\{F^{\text{expert}_i}\}_{i=1\ldots n}$ and a gating network $G$ (see Figure 2). From each expert, a feature extraction layer $\ell$ is selected, and the corresponding feature maps $f_\ell^{\text{expert}_i}$ are concatenated and provided as input to the gate. Given an input image $x \in \mathcal{I}^{H\times W\times C}$, where $H$ and $W$ denote the spatial dimensions, $C = 3$ the number of channels, and $\mathcal{I} = [0, 1]$, the gating network predicts a set of expert-specific weights

$$\left(w^{\text{expert}_1}, \ldots, w^{\text{expert}_n}\right) = G\left(f_\ell^{\text{expert}_i}\big|_{i=1,\ldots,n}\right). \quad (1)$$

In classical MoE models, the gating network replaces the fixed averaging of expert predictions used in standard ensembles by predicting dynamic, input-dependent weights that control each expert's contribution. Expert outputs are then multiplied by these weights rather than averaged as in an ensemble [1], [6], [7]. However, object detectors produce structured predictions that jointly encode BBox regression, objectness, and class probabilities. Consequently, ensembles of object detectors do not perform explicit averaging. Instead, they concatenate predictions and rely on post-processing, such as NMS. To transfer the MoE principle to object detection, we therefore apply the gate to weight all expert predictions, including BBox coordinates and class scores, before decoding the predictions and applying a final fusion step such as NMS suppression. This design choice is necessary because weighting only BBox coordinates or only class scores would break the tight coupling between localization and classification in object detectors, leading to inconsistent detections and unreliable postprocessing.

Let $y^{\text{expert}_i}$ denote the raw output of expert $i$, comprising all predicted BBox coordinates and associated scores. Given the expert weights $w^{\text{expert}_i}$ predicted by the gate, the weighted detector outputs are computed as

$$\tilde{y}^{\text{expert}_i} = w^{\text{expert}_i} \cdot y^{\text{expert}_i}, \quad i = 1, \ldots, n. \quad (2)$$

The weighted predictions are then decoded using a shared anchor configuration to recover BBox coordinates in image space. Finally, the decoded predictions from all experts are merged using a post-processing step, such as NMS, to obtain the final set of detections.

### B. MoE Training

For MoE training, the expert weights are frozen, and only the gate is trained with the same loss as used for single models.

**Balancing loss**: To prevent expert collapse and encourage balanced expert utilization, several existing auxiliary balancing losses are used. For this, expert importance is defined as the cumulative contribution of each expert across a training batch as measured by the gating mechanism [33]. For a given expert, this quantity is obtained by summing the gating weights assigned to that expert over all samples in the batch. Experts that receive consistently large gating weights are therefore considered more important under this definition.

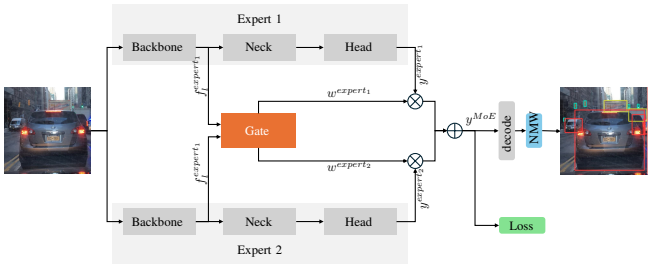



Fig. 2: Proposed MoE architecture consisting of two expert models.

The importance loss [33] is designed to mitigate imbalanced expert utilization by penalizing large disparities in expert importance across the batch. It measures how unevenly the total gating mass is distributed among experts and applies a stronger penalty as the variation between experts increases. By minimizing this objective, the model is encouraged to use experts more evenly.

A probabilistic alternative is the KL divergence loss [34], which treats expert importance as a discrete probability distribution over experts. The importance values are first normalized such that they sum to one across all experts. The loss then measures the divergence between this normalized distribution and a uniform distribution. Minimizing this divergence ensures that each expert receives an equal share of the total routing probability at the batch level.

Entropy-based loss [35] follows a similar motivation but directly maximizes the entropy of the normalized importance distribution. Higher entropy corresponds to a more uniform allocation of routing probability across experts.

This objective promotes balanced expert usage when aggregated over the batch but does not explicitly constrain the sharpness of routing decisions for individual samples. To address this limitation, a sample-wise entropy regularization term is applied before batch aggregation. Instead of operating on batch-level importance, this loss penalizes low-entropy gating distributions at the sample level. As a result, the gating network is discouraged from making overly confident routing decisions for individual inputs. The final training objective combines the task loss with this entropy regularization term, weighted by a coefficient $\lambda$.

**Domain-aware gate training:** The standard formulation uses the object detection loss, so the gate is not forced to route inputs to the experts, which correspond to the domain to which the input belongs. Instead, the MoE training aims to minimize object detection error. Domain-aware gate training uses cross-entropy loss on the input-domain labels, thereby enforcing routing inputs according to their domain.

## IV. Experimental Setup

We use an MoE containing two experts, trained on semantically disjoint subsets of data, inspired by our previous works [7], [32]. We used YOLO [10]-based detectors as expert models because their one-stage design, anchor-based prediction mechanism, and efficient inference pipeline make them a widely adopted and representative class of modern object detectors for evaluating model-level MoEs.

### A. Dataset Split

We use the Berkeley DeepDrive 100K (BDD100K) dataset [11], which we split according to the *time of day* metadata parameter (see Figures 3 and 4). We use *nighttime* data to train the *nighttime* expert model, and *daytime* data to train the *daytime* expert model. The BDD100K dataset contains approximately 35K daytime and 27K nighttime images. 27,971 images are randomly selected to train each expert, ensuring balanced training and eliminating bias. Given the absence of labeled test data in the BDD100K dataset, the training images are split at 90:10, resulting in 25,174 training and 2,798 validation images.

The original validation data is used as test data, with 3,929 test images per subset. In addition to *daytime* and *nighttime* data, we report results on 3,939 *dawn/dusk* images, and 1,876 *undefined* images.

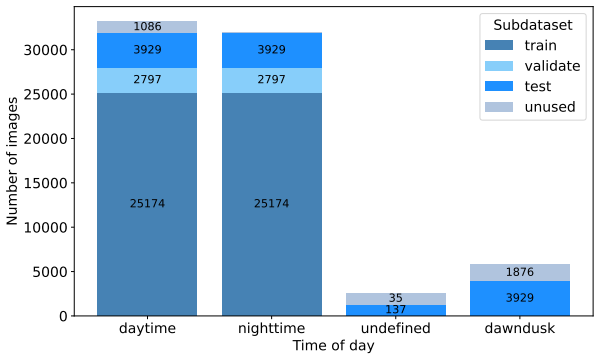



Fig. 3: Distribution of BDD100K data according to the metadata parameter *timeofday*.

### B. Training Settings

We used open-source PyTorch implementations of YOLOv7 [36][1] and YOLOv9 [37][2]. For each version, one large and one medium-sized variant were chosen. Models were trained for 200 epochs and evaluated with a confidence threshold of 0.001 and an IoU threshold of 0.6 at a NVIDIA RTX 2080 Ti GPU with 11GB VRAM.

Each expert model was trained on the corresponding subset. The baseline and the MoE were trained on the combined *daytime-nighttime* train data. During MoE training, expert weights were frozen. We report mAP50 on the test data for all models.

[1]https://github.com/WongKinYiu/yolov7
[2]https://github.com/WongKinYiu/yolov9


Fig. 4: Exemplary images with different values of the *timeofday* metadata parameter in the BDD100K dataset

The anchor configurations for all YOLOv7 models were optimized on the combined training data prior to training. As these anchors are also used for scaling during postprocessing, we use identical anchors for all models, optimized on the combined domains. For YOLOv9 inference, the primary detection head was used.

For the entropy balancing loss in an MoE, the regulation weight $\lambda$ was set to 0.1 for YOLOv7 and 0.3 for YOLOv9.

## V. Evaluation

### A. Detection Performance

**Performance of single models:** We have compared the detection performance of single models, ensembles, and an MoE (see Table I). Expert models perform the best on the corresponding subset they were trained on. In most cases, baselines outperform the expert models.

**Ensemble variants:** An ensemble of two experts was used as a comparison approach for the proposed MoE. For an ensemble, we have compared several methods for fusing the outputs of individual models (see Table II). For SoftNMS [16], $\sigma = 0.5$ and $\tau = 0.3$ were used. NMW [15] showed the best mAP for all model variants. Across all methods, pre-weighting BBox confidence scores before applying them based on each model's relative mAP further increased mAP. Therefore, NMW with mAP pre-weighting was used for all ensemble models.

**MoE performance:** The results (see Table I) show that MoEs consistently outperform fixed ensembles across model variants, indicating that learned routing is more effective than static fusion of expert predictions. However, MoEs generally do not surpass the strongest baseline models trained on the full, joint dataset, which benefit from exposure to the complete data distribution. Table III summarizes the relative mAP improvement compared to single experts and ensembles. The largest relative improvement could be reached on in-domain nighttime and daytime data.

TABLE I: Model comparison over data subsets.

| Model | mAP ↑ (%) on data subsets | | | | | | Num. params, M | Inf. time, ms |
|---|---|---|---|---|---|---|---|---|
| | Day | Night | Dawn/ Dusk | Undef. | Day+ Night | All | | |
| **YOLOv7-tiny, trained from scratch** | | | | | | | | |
| Baseline | 46.39 | **45.80** | **45.87** | **69.30** | **46.30** | **46.16** | 6.2 | 82 |
| Daytime expert | **46.53** | 34.92 | 45.00 | 56.33 | 42.78 | 43.50 | 6.2 | 82 |
| Nighttime expert | 32.62 | 43.20 | 34.62 | 59.60 | 36.55 | 35.86 | 6.2 | 82 |
| Ensemble | 45.97 | 42.67 | 45.18 | 61.57 | 44.90 | 45.10 | 12.4 | 186 |
| MoE | 46.54 | 43.82 | 45.52 | 60.36 | 45.97 | 45.79 | 12.4 | 82 |
| **YOLOv7, pre-trained on COCO** | | | | | | | | |
| Baseline | 59.68 | **56.51** | **59.31** | 78.65 | **58.42** | **58.70** | 63.9 | 73 |
| Daytime expert | **60.07** | 49.46 | 58.00 | **76.78** | 56.39 | 56.95 | 63.9 | 73 |
| Nighttime expert | 49.16 | 54.42 | 50.96 | 72.37 | 50.73 | 50.80 | 63.9 | 73 |
| Ensemble NMW | 59.65 | 54.38 | 58.34 | 78.05 | 57.52 | 57.90 | 127.8 | 157 |
| MoE | **60.07** | 54.75 | 58.40 | 77.24 | 58.11 | 58.21 | 127.8 | 80 |
| **YOLOv7x, pre-trained on COCO** | | | | | | | | |
| Baseline | **60.82** | **57.08** | **60.08** | **78.78** | **59.45** | **59.66** | 71.3 | 82 |
| Daytime expert | 60.47 | 50.40 | 58.94 | 76.26 | 57.07 | 57.66 | 71.3 | 82 |
| Nighttime expert | 51.26 | 54.79 | 52.30 | 67.81 | 52.27 | 52.21 | 71.3 | 82 |
| Ensemble | 60.54 | 54.26 | 59.48 | 77.80 | 58.36 | 58.69 | 142.6 | 170 |
| MoE | **60.82** | 55.04 | 59.39 | 74.38 | 58.75 | 58.95 | 142.6 | 97 |
| **YOLOv7x, trained from scratch** | | | | | | | | |
| Baseline | **59.79** | **56.06** | **59.18** | **76.84** | **58.43** | **58.64** | 71.3 | 83 |
| Daytime expert | 59.49 | 49.28 | 57.73 | 74.29 | 56.11 | 56.64 | 71.3 | 83 |
| Nighttime expert | 48.34 | 53.34 | 49.95 | 68.33 | 49.83 | 49.83 | 71.3 | 83 |
| Ensemble | 59.09 | 53.45 | 58.32 | 76.39 | 57.12 | 57.55 | 142.6 | 178 |
| MoE | 59.52 | 53.86 | 57.96 | 76.01 | 57.60 | 57.68 | 142.6 | 96 |
| **YOLOv9-S, pre-trained on COCO** | | | | | | | | |
| Baseline | **51.43** | **50.26** | **51.33** | **66.83** | **51.04** | **51.08** | 7.1 | 54 |
| Daytime expert | 51.15 | 40.00 | 49.93 | 66.03 | 47.53 | 48.27 | 7.1 | 54 |
| Nighttime expert | 38.76 | 46.28 | 40.54 | 49.24 | 41.45 | 41.04 | 7.1 | 54 |
| Ensemble | 50.14 | 45.91 | 49.79 | 68.27 | 48.82 | 49.09 | 14.2 | 73 |
| MoE | 51.31 | 47.20 | 50.71 | 64.83 | 50.16 | 50.24 | 14.2 | 76 |
| **YOLOv9c, pre-trained on COCO** | | | | | | | | |
| Baseline | 56.83 | **54.38** | **56.69** | **73.52** | **55.91** | **56.15** | 25.3 | 63 |
| Daytime expert | **56.95** | 49.70 | 55.53 | 69.37 | 54.41 | 54.53 | 25.3 | 63 |
| Nighttime expert | 47.22 | 52.07 | 49.25 | 64.49 | 48.80 | 48.86 | 25.3 | 63 |
| Ensemble | 56.50 | 52.66 | 56.08 | 74.44 | 55.07 | 55.29 | 50.6 | 98 |
| MoE | 57.75 | 53.17 | 56.89 | 73.73 | 55.85 | 56.13 | 50.6 | 94 |
| **YOLOv9c, trained from scratch** | | | | | | | | |
| Baseline | **56.68** | **53.49** | **55.97** | **71.23** | **55.65** | **55.72** | 25.3 | 63 |
| Daytime expert | 56.03 | 45.74 | 54.40 | 69.04 | 52.71 | 53.21 | 25.3 | 63 |
| Nighttime expert | 44.99 | 50.52 | 46.45 | 67.71 | 46.85 | 46.65 | 25.3 | 63 |
| Ensemble | 55.61 | 50.95 | 54.89 | 73.73 | 54.12 | 54.34 | 50.6 | 98 |
| MoE | 56.24 | 51.25 | 55.50 | 71.72 | 54.66 | 54.85 | 50.6 | 93 |
| **YOLOv9-E, pre-trained on COCO** | | | | | | | | |
| Baseline | 59.16 | **56.02** | **59.04** | 73.61 | **58.01** | **58.27** | 57.3 | 69 |
| Daytime expert | **59.48** | 51.23 | 60.50 | **74.37** | 56.61 | 57.89 | 57.3 | 69 |
| Nighttime expert | 50.28 | 53.35 | 51.59 | 68.48 | 51.11 | 51.18 | 57.3 | 69 |
| Ensemble | 59.21 | 54.49 | 60.67 | 77.12 | 57.43 | 58.50 | 114.6 | 107 |
| MoE | 59.93 | 55.36 | 61.41 | 75.24 | 58.12 | 59.19 | 114.6 | 107 |

TABLE II: Output fusion approaches, evaluated for a YOLOv7x ensemble.

| Algorithm | mAP ↑ (%) | |
|---|---|---|
| | Standard | With mAP re-weighting |
| NMS | 58.29 | 58.53 |
| WBF [14] | 56.09 | 55.94 |
| NMW [15] | 58.45 | **58.69** |
| SoftNMS [16] | 41.16 | 42.38 |

**Performance over data subdomains:** The domain-specific experts achieve their strongest performance on the subsets they were trained on (daytime or nighttime) but degrade noticeably on dawn/dusk and undefined images, indicating limited generalization beyond their specialization. We consider the dawn/dusk and undefined subsets as out-of-distribution (OOD) data since they are not included in the expert training splits and exhibit systematically lower detection performance across all models, indicating domain characteristics that differ from the training distribution. The baseline models are more robust on these OOD subsets and generally outperform individual experts, reflecting their broader exposure during training. Standard ensembles improve robustness over single experts by aggregating predictions, but their gains on OOD data remain limited (see Table I). The MoE consistently matches or slightly exceeds ensemble performance on these subsets and, in several configurations, approaches the baseline performance.

TABLE III: Relative mAP improvement of YOLOv7x-based MoE compared to other models.

| | Day | Night | Dawn/Dusk | Undef. | All |
|---|---|---|---|---|---|
| Compared to experts | **0.1565** | 0.1004 | 0.1281 | 0.0990 | 0.1279 |
| Compared to ensemble | 0.0111 | **0.0164** | 0.0106 | 0.0094 | 0.0127 |

TABLE IV: MoE ablation studies for YOLOv7x.

| Model | mAP ↑ (%) on data subsets | | | | | |
|---|---|---|---|---|---|---|
| | Day | Night | Dawn/ Dusk | Undef. | Day+ Night | All |
| **No domain-aware routing, different balancing losses** | | | | | | |
| No balancing loss | 60.47 | 54.20 | 58.95 | **75.81** | 58.34 | 58.48 |
| Importance loss | 60.42 | 54.68 | 58.95 | 70.26 | 58.42 | 58.51 |
| KL loss | 60.47 | 54.20 | 58.95 | **75.81** | 58.34 | 58.48 |
| Batch-wise entropy loss | 60.45 | 54.79 | 58.95 | 69.65 | 58.46 | 58.53 |
| Sample-wise entropy loss | **60.82** | **55.04** | **59.39** | 74.38 | **58.75** | **58.95** |
| **Domain-aware gate training, different balancing losses** | | | | | | |
| No balancing loss | 60.45 | **54.95** | 59.31 | **75.79** | 58.50 | **58.71** |
| Importance loss | 60.32 | 54.90 | 59.34 | 70.02 | 58.44 | 58.68 |
| KL loss loss | **60.49** | **54.95** | 59.29 | 70.63 | **58.52** | 58.70 |
| Batch-wise entropy loss | 60.47 | 54.86 | 59.24 | 73.90 | 58.49 | 58.67 |
| Sample-wise entropy loss | 60.46 | 54.90 | **59.33** | 75.75 | 58.49 | **58.71** |
| **Different gate architectures** | | | | | | |
| 1 FC layer | **60.83** | 54.93 | 59.31 | 70.01 | 58.71 | 58.79 |
| 2 FC layers | 60.48 | 54.97 | **59.40** | **75.75** | 58.53 | 58.76 |
| Conv and 2 FC layers | 60.82 | **55.04** | 59.39 | 74.38 | **58.75** | **58.95** |
| 2 Conv and 2 FC layers | 60.80 | 55.14 | 59.37 | 77.14 | 58.74 | 58.88 |
| **Different number of gates** | | | | | | |
| Single | **60.82** | **55.04** | **59.39** | 74.38 | **58.75** | **58.95** |
| Spatial | 60.53 | 55.13 | 59.32 | 76.94 | 58.67 | 58.87 |
| Class-wise | 60.37 | 53.29 | 59.14 | 76.28 | 58.01 | 58.38 |
| **Different feature extraction layers** | | | | | | |
| Early layer (L43) | **60.83** | **55.11** | 59.35 | **76.46** | **58.74** | **58.86** |
| Middle layer (L58) | 60.82 | 55.04 | 59.39 | 74.38 | 58.75 | 58.95 |
| Late layer (L117) | 60.55 | 54.90 | **59.51** | 70.30 | 58.54 | 58.78 |
| **Fixed weights instead of routing** | | | | | | |
| Equal-weighted | 57.34 | 54.14 | 57.75 | 73.31 | 56.20 | 56.69 |
| mAP-weighted | 57.74 | 53.90 | 58.17 | 75.00 | 56.41 | 56.98 |
| Routing | **60.82** | **55.04** | **59.39** | **74.38** | **58.75** | **58.95** |

### *B. Ablation studies*

**Training from scratch:** Model pre-trained on COCO [38] showed higher mAP than models trained from scratch on the corresponding BDD100K subset (see Table I).

**Balancing loss:** We performed a grid search to optimize the scale of each loss and a decay factor relative to the training epoch. Among the evaluated losses, sample-wise entropy achieved the highest overall mAP. Functions were evaluated to prevent the MoE model from collapsing to a single expert, an effect observed in the ablation studies (see Table IV).

**Domain-aware training:** Using cross-entropy loss on the input-domain labels has performed worse than unsupervised training over all losses, but still better than a single expert. A possible reason is that enforcing domain-based routing restricts the gate to coarse metadata labels and prevents it from exploiting finer-grained visual cues that are more informative for optimizing detection performance.

**Feature extraction layer:** Our gate uses features from the last layer of the backbone (the feature pyramid) as input from experts. It is referred to as a middle layer (see Table IV). In YOLO models, it corresponds to the highest level of abstraction and the lowest feature size. This makes it well-suited for a global, image-level gating decision. We also evaluated two

alternative locations in YOLOv7: an earlier backbone layer (L58), providing features closer to the image, and the final neck layer (L117). The neck refines backbone features by upsampling and downsampling across pyramid levels while using skip connections to combine multi-scale information. Its final layer also has high abstraction and low feature count. Among all tested configurations, using the last backbone layer achieved the highest mAP.

**Gate architecture:** We evaluated four gate architectures with increasing complexity:

- One FC layer: AvgPool → FC(2) → Softmax
- Two FC layers: AvgPool → FC(512) → FC(2) → Softmax
- One conv and two FC layers: Conv(1×1) → AvgPool → FC(512) → FC(2) → Softmax
- Two conv and two FC layers: Conv(1×1) → BN → Conv(3×3) → BN → AvgPool → FC(512) → FC(2) → Softmax

The variant with one convolutional and two fully connected layers performed best (see Table IV). This architecture evidently provides sufficient representational capacity while remaining simple enough to avoid overfitting.

**Number of gates:** In addition to a single image-level gate, we also evaluated a classwise gate that consists of one simple gate per class and a shared gate for coordinates and objectness, allowing class scores to be merged independently of box coordinates. In addition, we evaluated a spatial gate with one gate per spatial feature location (e.g., 20×20 in YOLOv7). These gate weights are bilinearly interpolated for each output layer and merge both coordinates and class scores for the corresponding image region. The single gate achieved the highest mAP and was therefore chosen for other experiments.

**Fixed weights:** Omitting the gates and using fixed expert weights instead of the predicted ones resulted in lower mAP (see Table IV), thus stressing the benefit of using MoEs.

### C. Impact on Interpretability

In this work, interpretability is defined as the ability to attribute detection outcomes to individual experts and to analyze how expert contributions vary across input conditions. For object detection, this includes understanding which expert dominates a prediction, how expert confidence changes across domains, and how disagreement between experts relates to final detection decisions. An interpretable model, therefore, exposes both global patterns of expert specialization across datasets and local decision behavior at the level of individual images and detected objects.

**Routing analysis:** The routing analysis provides a global view of expert specialization by examining the distribution of expert weights predicted by the gating network across data subsets. Figure 5 demonstrates that the gate learns domain-dependent routing behavior, even when no domain-aware training is used. In particular, it consistently routes inputs to the expert trained on the corresponding domain. For daytime images, the gate assigns high weights to the daytime expert and low weights to the nighttime expert, while the opposite

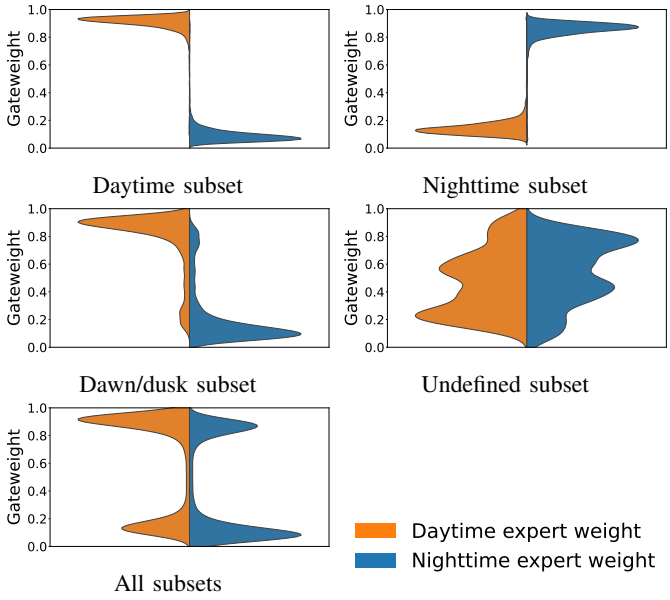



Fig. 5: Distribution of expert weights, predicted by a simple gate in a YOLOv7x-based MoE with samplewise entropy balancing loss and no domain-aware training.

behavior is observed for nighttime images, indicating domain-consistent expert weighting. For dawn/dusk images, the routing is skewed toward the daytime expert, suggesting that visual characteristics of this transitional domain are closer to daytime. In contrast, images from the undefined subset exhibit a broad and balanced distribution of gate values across both experts, indicating increased uncertainty and a more even reliance on both models for OOD data.

This observed routing behavior directly enhances interpretability compared to standard ensembles. In ensemble models, predictions are merged through fixed or heuristic post-processing, making it difficult to attribute a final detection to a specific expert or to analyze expert preference under changing input conditions. In contrast, the MoE explicitly exposes expert contributions through input-dependent gating weights. These weights provide a transparent and quantitative explanation of how much each expert influences the final prediction for a given image. As a result, expert specialization, domain ambiguity, and uncertainty become observable properties of the model rather than implicit effects hidden by postprocessing.

**Disagreement analysis:** The disagreement analysis focuses on local interpretability by examining how expert predictions differ at the object level. Quantitative results (see Table V) show that expert agreement dominates across all subsets, while a substantial number of BBoxes are predicted by only one expert. Daytime images exhibit a higher number of detections exclusive to the daytime expert, while nighttime images show a corresponding increase in detections exclusive to the nighttime expert. For dawn/dusk, and undefined images, the number of exclusive detections remains high for both experts, indicating increased ambiguity and reduced domain specificity.

Qualitative examples (see Figure 6) visualize expert agree-

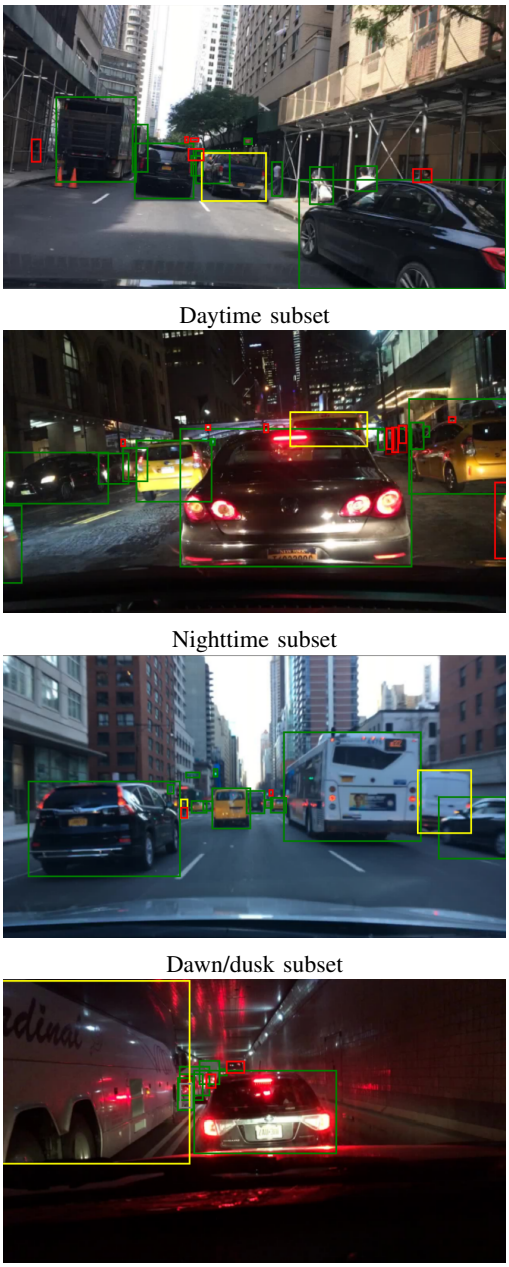
TABLE V: Average number of BBoxes per image for each disagreement case.

| Data subset | Full expert agreement | Expert agreement on label | Detection only by nighttime expert | Detection only by daytime expert |
|---|---|---|---|---|
| Daytime | **16.12** | 0.29 | 2.97 | **5.63** |
| Nighttime | 14.78 | 0.09 | **3.29** | 3.90 |
| Dawn/Dusk | 15.95 | **0.21** | 3.00 | 4.69 |
| Undefined | 7.81 | 0.07 | 1.49 | 2.28 |
| All | 15.54 | 0.20 | 3.07 | 4.72 |

ment and disagreement by indicating for each BBox whether both experts detect the object, disagree on its label, or fail to detect it. Across all subsets, BBoxes with full expert agreement dominate, while a smaller number of objects exhibit label disagreement. Importantly, the examples also show objects that are detected by only one expert (red BBoxes), highlighting missed detections by the other expert. This pattern is visible in all subsets and reflects complementary expert behavior rather than systematic conflicts. By explicitly exposing which expert contributes to each detection and where detections are missing, the MoE enables tracing final predictions back to individual expert decisions. In contrast to standard ensembles, which merge predictions without preserving such attribution, the MoE provides a transparent view of expert agreement, disagreement, and failure cases at the object level.

## VI. Conclusion

In this work, we studied model-level mixtures of experts for object detection and proposed an architecture that combines multiple YOLO-based detectors via a learned gating mechanism applied prior to detection postprocessing. Beyond improving detection performance, the proposed approach explicitly enhances interpretability by exposing input-dependent expert contributions and enabling attribution of detection outcomes to individual experts. Through routing and disagreement analyses, we showed that the model learns meaningful expert specialization across domains and provides insight into how conflicting expert predictions are resolved at both the image and object levels. Experimental results on a semantically split detection task demonstrate consistent improvements over standard ensembling and competitive performance relative to single model baselines, while offering a transparent alternative to heuristic ensemble fusion. The proposed method benefits from modular expert specialization and explicit decision weighting but introduces additional complexity in prediction fusion and relies on meaningful domain separation to realize its advantages. Future work can explore scaling to larger numbers of experts, extending the approach to other detection architectures, and developing more robust gating strategies that explicitly account for uncertainty and sample level ambiguity.



Daytime subset

Nighttime subset

Dawn/dusk subset

Undefined subset

Fig. 6: Disagreement analysis helps trace MoE decisions back to expert decisions. BBoxes with full expert agreement are green, BBoxes of expert disagreement on label are yellow, and BBoxes of objects not detected by either expert are red. The results are for the YOLOv7x-based MoE with sample-wise entropy-balancing loss and no domain-aware training.

## ACKNOWLEDGMENT

This work was supported by funding from the Topic Engineering Secure Systems of the Helmholtz Association (HGF) and by KASTEL Security Research Labs (46.23.03).

## REFERENCES

[1] R. A. Jacobs, M. I. Jordan, S. J. Nowlan, and G. E. Hinton, "Adaptive mixtures of local experts," *Neural computation*, vol. 3, no. 1, 1991.

[2] W. Fedus, B. Zoph, and N. Shazeer, "Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity," *CoRR*, vol. abs/2101.03961, 2021.

[3] D. Dai, C. Deng, C. Zhao, R. Xu, H. Gao, D. Chen, J. Li, W. Zeng, X. Yu, Y. Wu *et al.*, "Deepseekmoe: Towards ultimate expert specialization in mixture-of-experts language models," *arXiv preprint arXiv:2401.06066*, 2024.

[4] C. Riquelme, J. Puigcerver, B. Mustafa, M. Neumann, R. Jenatton, A. S. Pinto, D. Keysers, and N. Houlsby, "Scaling vision with sparse mixture of experts," in *Advances in Neural Information Processing Systems (NIPS)*, 2021.

[5] H. Lee, Y. Xu, D. Chen, O. Firat, Y. Huang, M. Krikun, N. Shazeer, and Z. Chen, "Gshard: Scaling giant models with conditional computation and automatic sharding," in *International Conference on Learning Representations (ICLR)*, 2021.

[6] A. Valada, A. Dhall, and W. Burgard, "Convoluted Mixture of Deep Experts for Robust Semantic Segmentation," in *International Conference on Intelligent Robots and Systems (IROS) - Workshops*, 2016.

[7] S. Pavlitskaya, C. Hubschneider, M. Weber, R. Moritz, F. Hüger, P. Schlicht, and J. M. Zöllner, "Using mixture of expert models to gain insights into semantic segmentation," in *Conference on Computer Vision and Pattern Recognition (CVPR) - Workshops*, 2020.

[8] K. Ahmed, M. H. Baig, and L. Torresani, "Network of experts for large-scale image categorization," in *European Conference on Computer Vision (ECCV)*. Springer, 2016.

[9] O. Mees, A. Eitel, and W. Burgard, "Choosing smartly: Adaptive multimodal fusion for object detection in changing environments," in *International Conference on Intelligent Robots and Systems (IROS)*, 2016.

[10] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[11] F. Yu, H. Chen, X. Wang, W. Xian, Y. Chen, F. Liu, V. Madhavan, and T. Darrell, "BDD100K: A diverse driving dataset for heterogeneous multitask learning," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[12] F. Doshi-Velez and B. Kim, "Towards a rigorous science of interpretable machine learning," *arXiv preprint arXiv:1702.08608*, 2017.

[13] C. Rudin, "Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead," *Nat. Mach. Intell.*, 2019.

[14] R. Solovyev, W. Wang, and T. Gabruseva, "Weighted boxes fusion: Ensembling boxes from different object detection models," *Image and Vision Computing*, 2021.

[15] H. Zhou, Z. Li, C. Ning, and J. Tang, "Cad: Scale invariant framework for real-time object detection," in *International Conference on Computer Vision (ICCV) - Workshops*, 2017.

[16] N. Bodla, B. Singh, R. Chellappa, and L. S. Davis, "Soft-nms–improving object detection with one line of code," in *International Conference on Computer Vision (ICCV)*, 2017.

[17] R. Xu, H. Lin, K. Lu, L. Cao, and Y. Liu, "A forest fire detection system based on ensemble learning," *Forests*, 2021.

[18] Á. Casado-García and J. Heras, "Ensemble methods for object detection," in *European Conference on Artificial Intelligence (ECAI)*, 2020.

[19] J. Li, J. Qian, and Y. Zheng, "Ensemble r-fcn for object detection," in *International Conference on Ubiquitous Information Technologies and Applications*, 2017.

[20] C. Peng, T. Xiao, Z. Li, Y. Jiang, X. Zhang, K. Jia, G. Yu, and J. Sun, "Megdet: A large mini-batch object detector," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[21] J. Guo and S. Gould, "Deep cnn ensemble with data augmentation for object detection," *arXiv preprint arXiv:1506.07224*, 2015.

[22] Z. Cai and N. Vasconcelos, "Cascade r-cnn: Delving into high quality object detection," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[23] J. Lee, S.-K. Lee, and S.-I. Yang, "An ensemble method of cnn models for object detection," in *International Conference on Information and Communication Technology Convergence (ICTC)*, 2018.

[24] N. D. Vo, K. Nguyen, T. V. Nguyen, and K. Nguyen, "Ensemble of deep object detectors for page object detection," in *International Conference on Ubiquitous Information Management and Communication*, 2018.

[25] C. Bahhar, A. Ksibi, M. Ayadi, M. M. Jamjoom, Z. Ullah, B. O. Soufiene, and H. Sakli, "Wildfire and smoke detection using staged yolo model and ensemble cnn," *Electronics*, vol. 12, 2023.

[26] Ultralytics, "YOLOv5: A state-of-the-art real-time object detection system," 2021.

[27] D. Zhang, "A yolo-based approach for fire and smoke detection in iot surveillance systems." *International Journal of Advanced Computer Science & Applications*, vol. 15, no. 1, 2024.

[28] P. Huayhongthong, S. Rerk-u suk, S. Booddee, P. Padungweang, and K. Warasup, "Incremental object detection using ensemble modeling and deep transfer learning," in *International Conference on Computing and Information Technology*, 2020.

[29] L. Zhang, S. Huang, W. Liu, and D. Tao, "Learning a mixture of granularity-specific experts for fine-grained categorization," in *International Conference on Computer Vision (ICCV)*, 2019.

[30] S. Yang, J. Wen, and B. Fang, "Fg-moe: Heterogeneous mixture of experts model for fine-grained visual classification," *Pattern Recognition*, p. 113050, 2026.

[31] K. Oksuz, S. Kuzucu, T. Joy, and P. K. Dokania, "Mocae: Mixture of calibrated experts significantly improves object detection," *Trans. Mach. Learn. Res.*, 2024.

[32] S. Pavlitskaya, C. Hubschneider, and M. Weber, "Evaluating mixture-of-experts architectures for network aggregation," in *Deep Neural Networks and Data for Automated Driving: Robustness, Uncertainty Quantification, and Insights Towards Safety*, 2022.

[33] N. Shazeer, A. Mirhoseini, K. Maziarz, A. Davis, Q. Le, G. Hinton, and J. Dean, "Outrageously large neural networks: The sparsely-gated mixture-of-experts layer," *arXiv preprint arXiv:1701.06538*, 2017.

[34] S. Pavlitska, C. Hubschneider, L. Struppek, and J. M. Zöllner, "Sparsely-gated mixture-of-expert layers for cnn interpretability," in *International Joint Conference on Neural Networks (IJCNN)*, 2023.

[35] S. Pavlitska, H. Fan, K. Ditschuneit, and J. M. Zöllner, "Robust experts: the effect of adversarial training on cnns with sparse mixture-of-experts layers," in *International Conference on Computer Vision (ICCV) - Workshops*, 2025.

[36] C. Wang, A. Bochkovskiy, and H. M. Liao, "Yolov7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[37] C. Wang, I. Yeh, and H. M. Liao, "Yolov9: Learning what you want to learn using programmable gradient information," in *European Conference on Computer Vision (ECCV)*, 2024.

[38] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *European Conference on Computer Vision (ECCV)*. Springer, 2014, pp. 740–755.