Layerwise and Epochwise Visualization of Errors in Object Detection

Jaival Bhuptani*, Samprith Kalakata*, Abhipal Sharma*
*Tandon School of Engineering, New York University
jnb9582@nyu.edu srk9068@nyu.edu as20410@nyu.edu

I. Abstract

Understanding how and why object detection models fail is crucial for building interpretable and robust systems. This project proposes an **interactive visualization framework** that links quantitative TIDE error analysis with qualitative layerwise Grad-CAM explanations, allowing users to explore **error progression across epochs and network layers**. Rather than treating errors as isolated snapshots, the system provides a temporal-spatial view of error evolution, enabling examination of misclassified samples, and inspection of attention maps at different layers. By integrating charts, image grids, and layerwise visualizations into a unified dashboard, this framework offers actionable insights for model debugging, interpretability, and improved understanding of how object detectors learn and misclassify over training.

Our goal is to develop an error evolution tracker across epochs and layers to hopefully understand systematic failure patterns.

II. Research Questions

- How do different TIDE error categories (e.g., false positives, false negatives, localization errors) evolve across network layers and training epochs?
- Do specific error types correspond to distinctive Grad-CAM attention patterns or visual signatures?
- Can recurring error patterns or "failure signatures" be identified and compared across checkpoints, architectures, and datasets?
- What insights can an interactive dashboard provide for understanding, exploring, and debugging these behaviors?

III. Related Works

Error analysis and interpretability in object detection have been extensively studied. Bolya et al. [1] introduced **TIDE**, which decomposes detection errors into categories such as classification, localization, duplicates, and background false positives, enabling fine-grained analysis across detectors like Faster R-CNN, YOLO, and DETR.

Visual interpretability methods, including **Grad-CAM** [2], **Grad-CAM++** [3], **Layer-CAM** [4], and **Score-CAM** [5], generate class-discriminative heatmaps to provide insights into model attention. However, these techniques typically focus on single snapshots and do not link attention maps with quantitative error decomposition over training.

Our framework addresses this gap by unifying **TIDE-based error analysis** with **layerwise and epochwise Grad-CAM visualizations** in an interactive dashboard. This allows users to explore how detection failures develop over training, inspect individual misclassifications, and gain both quantitative and qualitative insights into model behavior.

IV. Methodology & Direction Overview

Our system visualizes how detection errors evolve across epochs and layers through a structured pipeline of six stages:

- **Step 1: Data Preparation** — Record TIDE error counts per epoch and store metadata for mispredicted validation images, including Grad-CAM maps and confidence metrics. The data is organized for filtering by epoch, layer, and error type.
- **Step 2: Global Error Evolution Dashboard** — Visualize how different TIDE error categories change across epochs using an interactive area or line chart with hover and filter capabilities.
- **Step 3: Error Category Selector** — Provide an interface (dropdown or buttons) for selecting a specific TIDE error type, dynamically updating subsequent visualizations.
- **Step 4: Misclassified Image Grid** — Display thumbnails of failed validation samples for the selected category and epoch, with hoverable metrics and clickable access to detailed analysis.
- **Step 5: Layerwise Grad-CAM Viewer** — Present Grad-CAM visualizations from multiple backbone layers side by side, enabling insight into how feature attention shifts across network depth.
- **Step 6: Epochwise Comparison** — For a particular image, allow users to explore how the image's final layer activation maps evolve throughout training via an epoch slider.

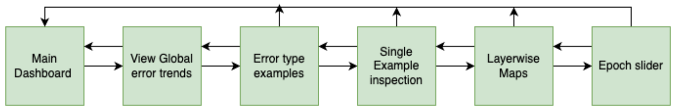

Fig. 1. User interaction flow for the visualization dashboard (Figure 1).

A. User Flow

The dashboard provides an intuitive interface for exploring model behavior across epochs and error types, as summarized in Figure 1. Users start at the **Main Dashboard**, then proceed to **View Global Error Trends** to inspect TIDE error distributions over training epochs. They can select a specific epoch and error category to filter results, which are displayed as **Error Type Examples** in a grid. Clicking on an image opens **Single Example Inspection**, showing detailed **Layerwise Maps** of Grad-CAM activations across backbone layers. Finally, an **Epoch Slider** allows temporal exploration of how the example's attention and predictions evolve throughout training. This workflow mirrors the system's six-step methodology, linking quantitative error trends with qualitative visual insights.

B. Datasets

We will use two datasets for evaluation. **COCO-2017** (train/val) contains over 118,000 training and 5,000 validation images across 80 categories, featuring complex scenes with multiple objects, occlusions, and varying sizes. **Pascal VOC 2012** (train/val) includes about 11,000 images across 20 categories, with simpler scenes and fewer objects per image, suitable for assessing model generalization on smaller datasets.

C. Implementation Stack

The interactive visualization system will be implemented using **Plotly Dash** to manage the dashboard and layout control, while **Plotly Express** or standard Plotly will be used for generating dynamic charts to visualize error evolution. **OpenCV** or **PIL** will handle overlaying Grad-CAM heatmaps and bounding boxes on validation images, and **Dash Bootstrap Components** can be optionally employed to create a clean, responsive interface with well-organized panels.

V. Timeline

TABLE I
PROJECT TIMELINE (OCT 24 – DEC 11)

Task	Description	Timeline
Data Preparation	Collect TIDE outputs, mispredictions, Grad-CAM maps, and relevant metadata for each epoch	Oct 24 – Oct 30
Global Error Evolution Dashboard	Implement visualization of TIDE error trends across epochs (stacked/multi-line charts)	Oct 31 – Nov 6
Error Category Selector	Add filtering controls for exploring specific TIDE error categories	Nov 7 – Nov 10
Misclassified Image Grid	Display failed validation examples with hover metrics and clickable image inspection	Nov 11 – Nov 17
Layerwise Grad-CAM Viewer	Visualize Grad-CAM maps from multiple layers with transition slider	Nov 18 – Nov 25
Epochwise Comparison	Enable per-image temporal visualization of Grad-CAM evolution across epochs	Nov 26 – Dec 2
Integration & Testing	Integrate modules into a unified dashboard and test user interactions	Dec 3 – Dec 7
Reporting & Documentation	Summarize findings and compile final project report	Dec 8 – Dec 11

VI. Expected Outcomes

We will develop an **interactive visualization system** that reveals how detection errors evolve across training epochs and network layers. The system will integrate quantitative **TIDE error statistics** with qualitative **Grad-CAM visualizations**, enabling users to trace the progression of specific error categories over time. Through an interactive dashboard, users will be able to filter by epoch, error type, and image instance to examine misclassified samples and observe how model attention shifts across layers. The final outcome will include a structured dataset linking TIDE categories with corresponding visual explanations, a **unified dashboard interface for exploration** of how models learn and misclassify throughout training.

VII. References

References

[1] Bolya, D., et al., "TIDE: A General Toolbox for Identifying Object Detection Errors," ECCV, 2020.
[2] Selvaraju, R. R., et al., "Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization," ICCV, 2017.
[3] Chattopadhyay, P., et al., "Grad-CAM++: Improved Visual Explanations for Deep Convolutional Networks," AAAI, 2018.
[4] Jiang, M., et al., "Layer-CAM: Exploring Hierarchical Class Activation Maps for Deep CNNs," CVPR, 2021.
[5] Wang, Y., et al., "Score-CAM: Score-Weighted Visual Explanations for CNNs," CVPR, 2020.