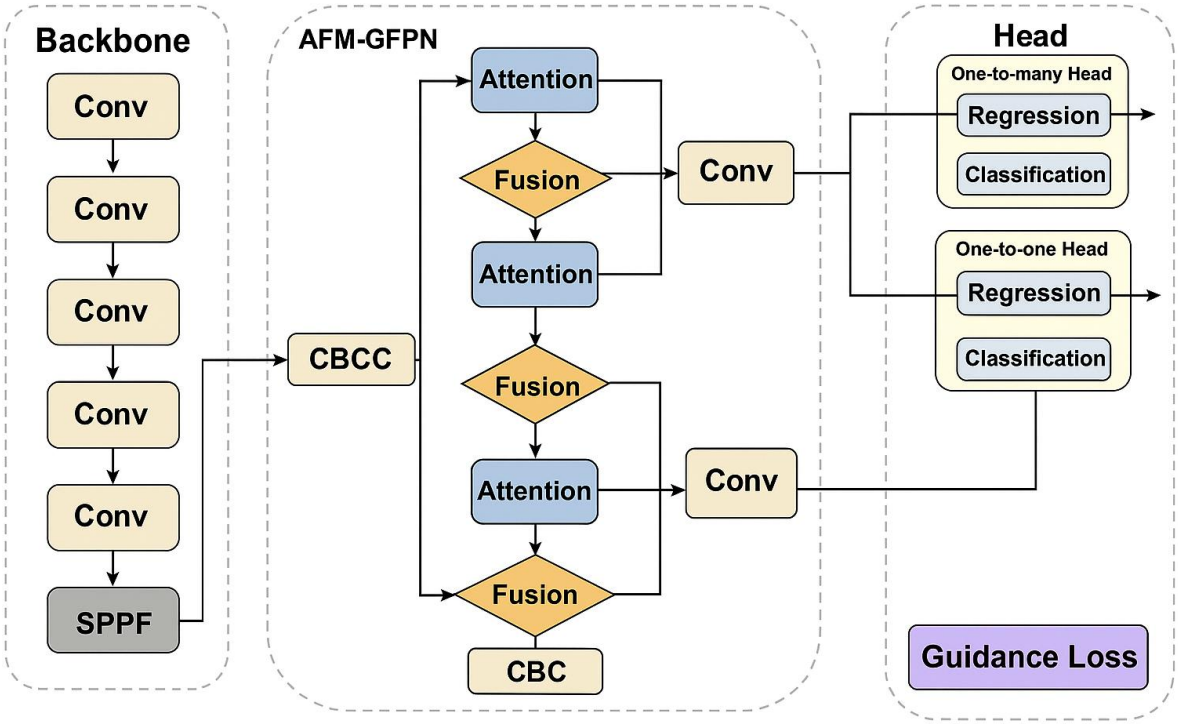


## ◇ 1. Backbone

The **feature extractor** – it converts the input image into low-resolution, high-channel representations.

**Components**:

- `Conv` × 4: Sequential convolutional blocks (likely Conv → BN → ReLU).
- `SPPF`: Spatial Pyramid Pooling – helps extract multi-scale context (important for small/large object detection).

**Purpose**:

- Detect multi-scale spatial features from raw image input.

## ◇ 2. AFM-GFPN (Attention-Fusion Multi-scale Global Feature Pyramid Network)

This replaces standard PANet/Neck layers from YOLO to fuse multi-scale features intelligently.

**Submodules**:

- **CBCC**: Possibly “Cross-Branch Channel Compression” – downscales and compresses features before fusion.
- **Attention** (Blue blocks): Applies **transformer-style** or **coordinate/spatial attention** to highlight useful features.
- **Fusion** (Orange diamonds): Merges features from different scales/resolutions.
- **CBC**: Likely a final **Channel Bottleneck Compression** block – minimizes redundancy.

**Key Design Idea**:

- Hierarchical attention after every fusion stage.
- Preserves both **global context** (for large patterns like zebra crossings) and **fine details** (like traffic lights).

## ◇ 3. Head

Performs final prediction for detection and classification.

**Heads Used**:

- **One-to-many Head**: For objects like vehicles or zebra crossings with **multiple instances**.
- **One-to-one Head**: For single-target critical detections (e.g. traffic light status).
- Each head has:
  - `Regression`: Predicts bounding boxes (x, y, w, h).
  - `Classification`: Predicts class label (e.g., red/green light, vehicle type, crosswalk).

## ◇ 4. Guidance Loss

Custom output block used instead of generic IoU/CIoU loss. Tailored to:

- Penalize **wrong instruction decisions** (e.g., "go" when a red light is present).
- Include **contextual consistency** between detected traffic light, zebra crossing, and vehicles.

# ☑How This Architecture Works for Visually Impaired Navigation:

| Requirement | How this model supports it |
|---|---|
| Detect small & distant traffic lights | Multi-scale fusion + attention (AFM-GFPN) |
| Handle overlapping vehicles | One-to-many head for dense object detection |
| Understand spatial layout (zebra) | Fusion layers + attention focus ground-level info |

| Requirement | How this model supports it |
|---|---|
| Give decision (go/stop/wait) | Custom loss + logic based on prediction branches |

## ⬅ END Summary

| Module | Role |
|---|---|
| **Backbone** | Extracts low- to high-level image features |
| **AFM-GFPN** | Enhances and merges features via attention |
| **Head** | Predicts bounding boxes + class per object |
| **Guidance Loss** | Guides model training to align with navigation logic |

## ☑ Full Flow

| Stage | Input Features | Output |
|---|---|---|
| `Conv` | RGB pixels | Shape/edge/color patterns |
| `SPPF` | All scales of features | Broad spatial understanding |
| `CBCC` | Compressed channel features | Balanced small & large object info |
| `Attention` | Focused areas (red light, zebra) | Suppressed background clutter |
| `Fusion` | Multi-level object clues | Unified representation |
| `Head` | Objects + boxes + classes | e.g., Red light on left, Car center |
| `Guidance Loss` | True vs. predicted decision | Punishes incorrect “Go” at red |