

From Strip Maps to Smart Dashboards: Digitizing Speed Zones with AI-Driven Methods

TexITE 2026 Meeting
Houston | April 8-10

Texas A&M Transportation Institute

Purpose

Speed zone management is a critical responsibility for transportation engineers and planners — yet the tools available have lagged far behind the complexity of the task. This project addresses that gap head-on.

- ❏ **Core Objective:** Provide engineers and planners with a modern visualization/analysis tool to more effectively and efficiently manage the district's speed zones — replacing outdated strip maps and fragmented file-sharing practices with a unified, data-driven dashboard.

The result is a smarter, faster, and more transparent approach to maintaining accurate speed zone records across the TxDOT Dallas District.



Challenges of Managing Speed Zone Maps

Communication Gaps

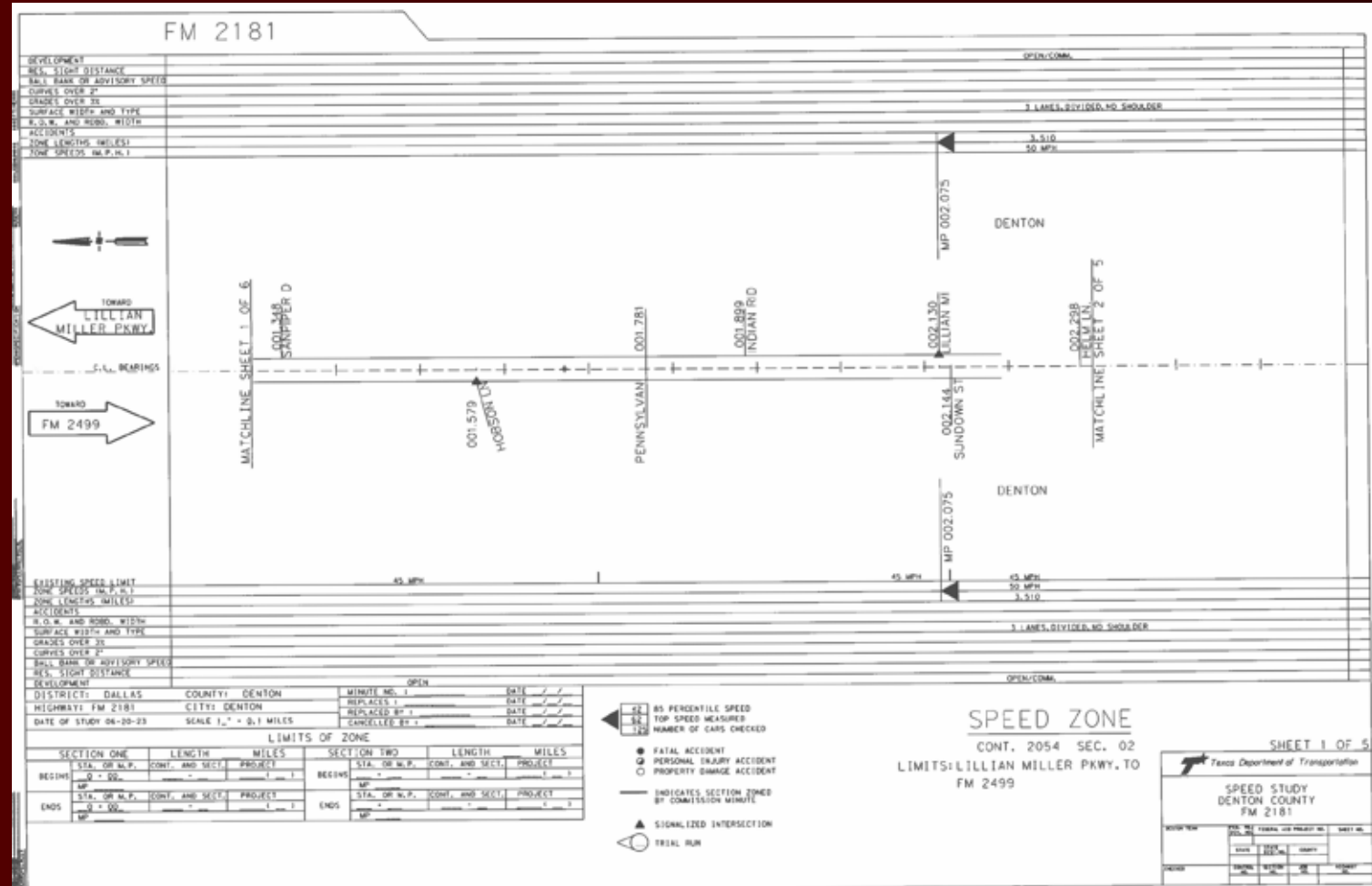
Responding to citizen requests and coordinating with local governments is difficult without an easily accessible, up-to-date visual reference that can be shared on demand.

Zone Limit and PSL Discrepancies

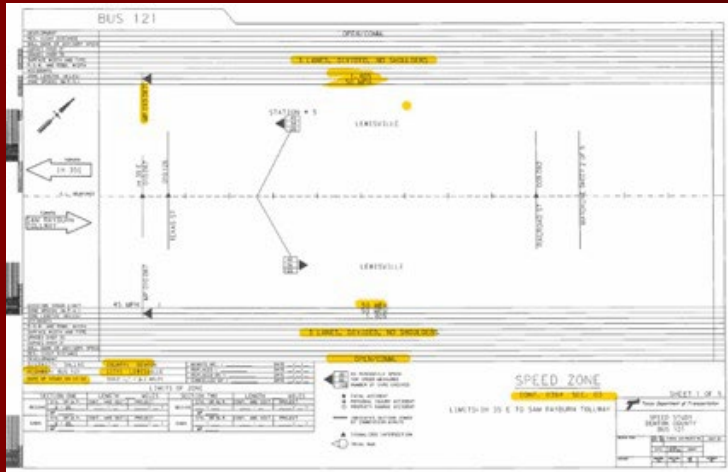
Identifying discrepancies between the Statewide Planning Map and strip maps, often due to highway realignments, mile point/DFO limits, or delayed implementation of approved PSLs by municipalities.

Inefficient File Sharing

Current distribution relies on hard copies or emailed electronic files — with no centralized repository — making it prone to outdated records and duplicated effort.



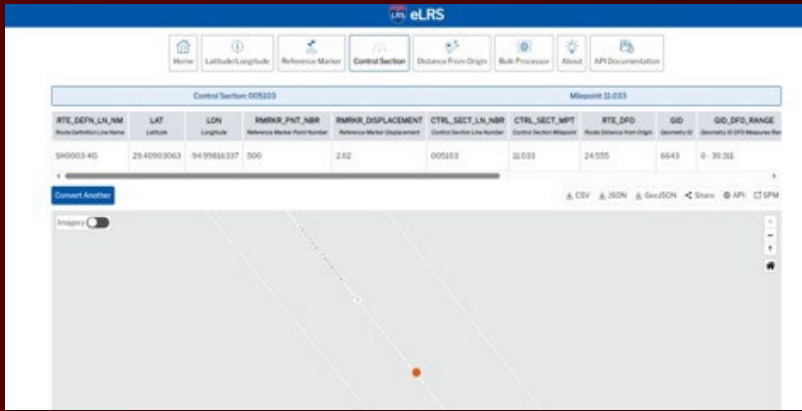
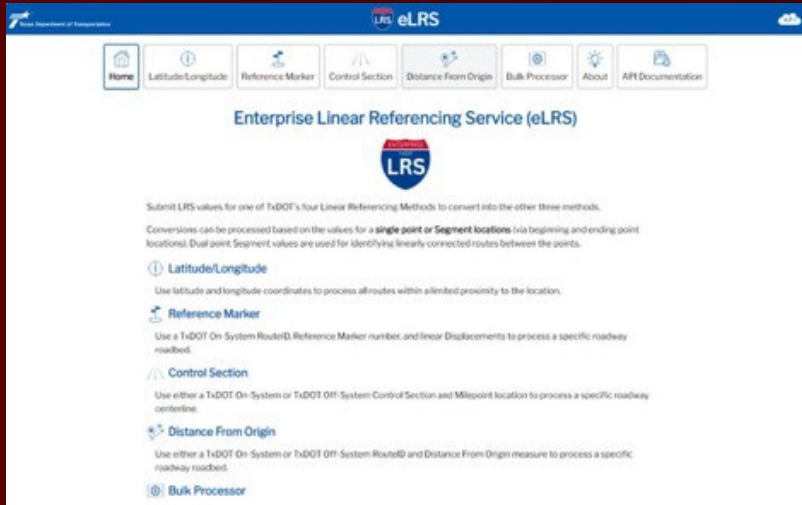
Automating Digitization Process



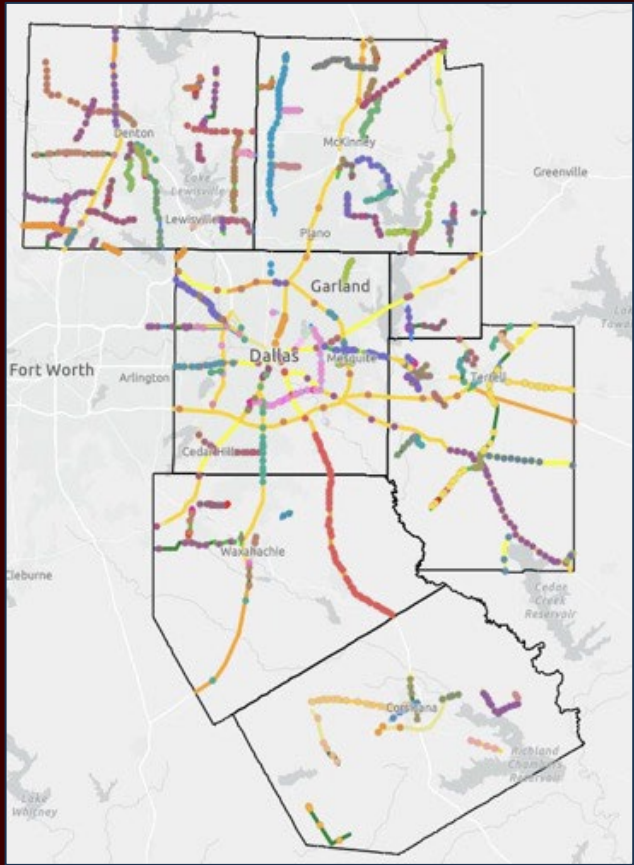
Step 1:
Developed Python script to extract segment information from strip map PDFs

```

pdf_extract_step_1
1. Import os
2. Import sys
3. Import Counter as C
4. From shapely.geometry import Polygon
5. Import re
6. Import pandas as pd
7. Import numpy as np
8. Import pykdtree as kdTree
9. From shapely.geometry import Point, Polygon, LineString
10. From shapely.geometry import LineString, Polygon, Point
11. From shapely.geometry import LineString, Polygon, Point
12. From shapely.geometry import LineString, Polygon, Point
13. From shapely.geometry import LineString, Polygon, Point
14. From shapely.geometry import LineString, Polygon, Point
15. From shapely.geometry import LineString, Polygon, Point
16. From shapely.geometry import LineString, Polygon, Point
17. From shapely.geometry import LineString, Polygon, Point
18. From shapely.geometry import LineString, Polygon, Point
19. From shapely.geometry import LineString, Polygon, Point
20. From shapely.geometry import LineString, Polygon, Point
21. From shapely.geometry import LineString, Polygon, Point
22. From shapely.geometry import LineString, Polygon, Point
23. From shapely.geometry import LineString, Polygon, Point
24. From shapely.geometry import LineString, Polygon, Point
25. From shapely.geometry import LineString, Polygon, Point
26. From shapely.geometry import LineString, Polygon, Point
27. From shapely.geometry import LineString, Polygon, Point
28. From shapely.geometry import LineString, Polygon, Point
29. From shapely.geometry import LineString, Polygon, Point
30. From shapely.geometry import LineString, Polygon, Point
31. From shapely.geometry import LineString, Polygon, Point
32. From shapely.geometry import LineString, Polygon, Point
33. From shapely.geometry import LineString, Polygon, Point
34. From shapely.geometry import LineString, Polygon, Point
35. From shapely.geometry import LineString, Polygon, Point
36. From shapely.geometry import LineString, Polygon, Point
37. From shapely.geometry import LineString, Polygon, Point
38. From shapely.geometry import LineString, Polygon, Point
39. From shapely.geometry import LineString, Polygon, Point
40. From shapely.geometry import LineString, Polygon, Point
41. From shapely.geometry import LineString, Polygon, Point
42. From shapely.geometry import LineString, Polygon, Point
43. From shapely.geometry import LineString, Polygon, Point
44. From shapely.geometry import LineString, Polygon, Point
45. From shapely.geometry import LineString, Polygon, Point
46. From shapely.geometry import LineString, Polygon, Point
47. From shapely.geometry import LineString, Polygon, Point
48. From shapely.geometry import LineString, Polygon, Point
49. From shapely.geometry import LineString, Polygon, Point
50. From shapely.geometry import LineString, Polygon, Point
51. From shapely.geometry import LineString, Polygon, Point
52. From shapely.geometry import LineString, Polygon, Point
53. From shapely.geometry import LineString, Polygon, Point
54. From shapely.geometry import LineString, Polygon, Point
55. From shapely.geometry import LineString, Polygon, Point
56. From shapely.geometry import LineString, Polygon, Point
57. From shapely.geometry import LineString, Polygon, Point
58. From shapely.geometry import LineString, Polygon, Point
59. From shapely.geometry import LineString, Polygon, Point
60. From shapely.geometry import LineString, Polygon, Point
61. From shapely.geometry import LineString, Polygon, Point
62. From shapely.geometry import LineString, Polygon, Point
63. From shapely.geometry import LineString, Polygon, Point
64. From shapely.geometry import LineString, Polygon, Point
65. From shapely.geometry import LineString, Polygon, Point
66. From shapely.geometry import LineString, Polygon, Point
67. From shapely.geometry import LineString, Polygon, Point
68. From shapely.geometry import LineString, Polygon, Point
69. From shapely.geometry import LineString, Polygon, Point
70. From shapely.geometry import LineString, Polygon, Point
71. From shapely.geometry import LineString, Polygon, Point
72. From shapely.geometry import LineString, Polygon, Point
73. From shapely.geometry import LineString, Polygon, Point
74. From shapely.geometry import LineString, Polygon, Point
75. From shapely.geometry import LineString, Polygon, Point
76. From shapely.geometry import LineString, Polygon, Point
77. From shapely.geometry import LineString, Polygon, Point
78. From shapely.geometry import LineString, Polygon, Point
79. From shapely.geometry import LineString, Polygon, Point
80. From shapely.geometry import LineString, Polygon, Point
81. From shapely.geometry import LineString, Polygon, Point
82. From shapely.geometry import LineString, Polygon, Point
83. From shapely.geometry import LineString, Polygon, Point
84. From shapely.geometry import LineString, Polygon, Point
85. From shapely.geometry import LineString, Polygon, Point
86. From shapely.geometry import LineString, Polygon, Point
87. From shapely.geometry import LineString, Polygon, Point
88. From shapely.geometry import LineString, Polygon, Point
89. From shapely.geometry import LineString, Polygon, Point
90. From shapely.geometry import LineString, Polygon, Point
91. From shapely.geometry import LineString, Polygon, Point
92. From shapely.geometry import LineString, Polygon, Point
93. From shapely.geometry import LineString, Polygon, Point
94. From shapely.geometry import LineString, Polygon, Point
95. From shapely.geometry import LineString, Polygon, Point
96. From shapely.geometry import LineString, Polygon, Point
97. From shapely.geometry import LineString, Polygon, Point
98. From shapely.geometry import LineString, Polygon, Point
99. From shapely.geometry import LineString, Polygon, Point
100. From shapely.geometry import LineString, Polygon, Point
    
```



Step 2: Using [TxDOT eLRS API](#) to geolocate segments & stations

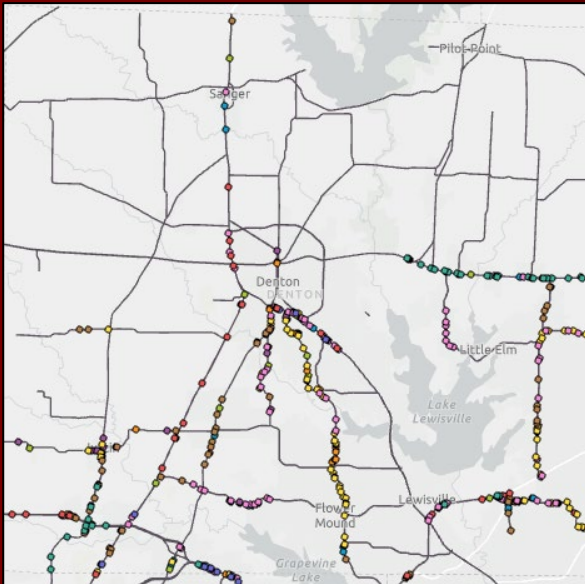


Step 3: Add segment/station to speed zone dashboard

Creating Posted Speed Limit (PSL) Layer

Step 1: Download crowdsource PSL data (Mapillary) and import into GIS.

- 4,525 data points (PSL, school zones, etc.)
- Filtered to ~300 points for dashboard speed zones
- 87.6% accuracy (matched with Google Earth imagery)

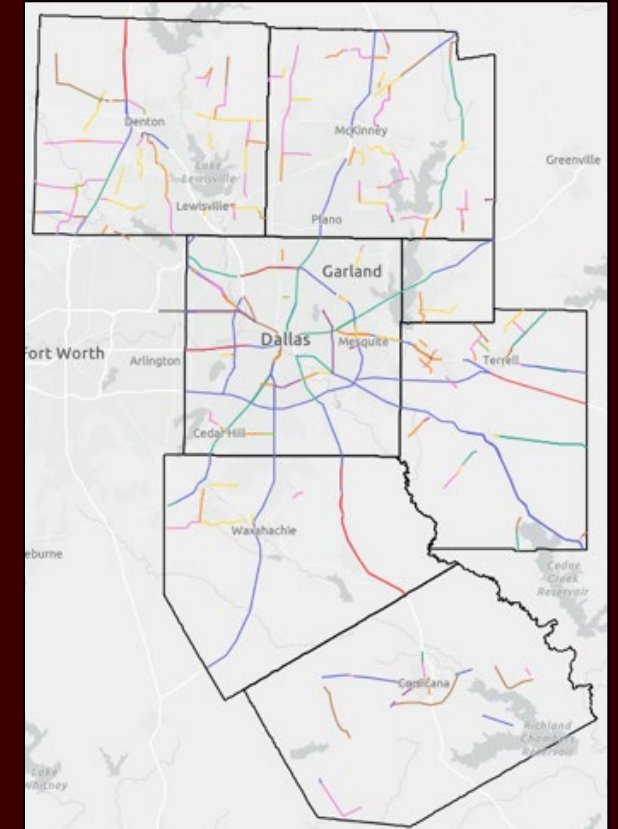


Step 2: Check Google Streetview to verify/adjust PSL at the begin/end segment (Manual).

Step 2a: Check Pathway video logs to verify/adjust PSL for entire segment (Vision Language Model).

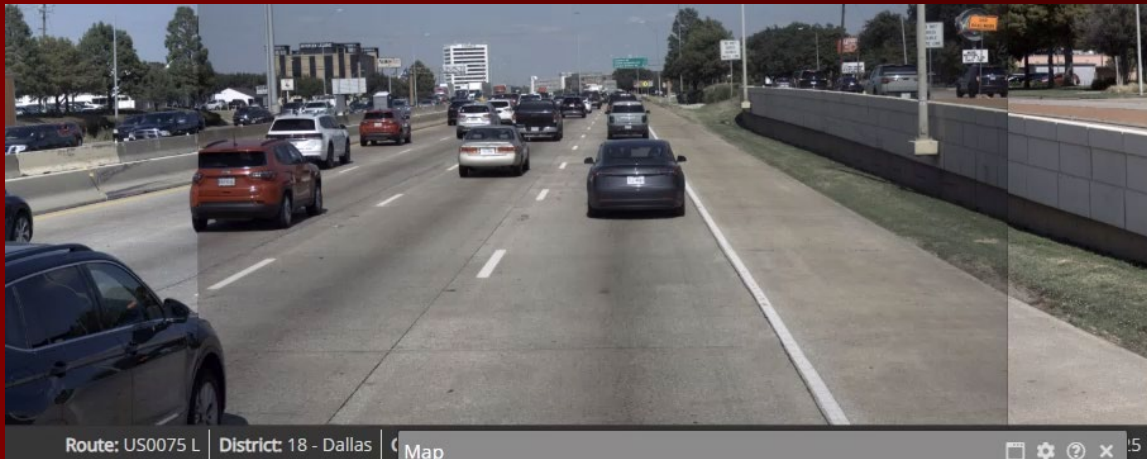
Step 2b: Check RIVAL dataset to verify/adjust PSL for entire segment (GIS) (Future Work)

Step 3: Add PSL layer to speed zone dashboard



Creating Posted Speed Limit (PSL) Layer

On-device Vision Language Model (VLM) Workflow for Posted Speed-Limit Identification

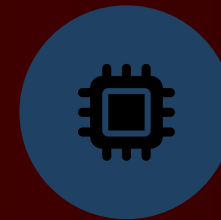


System architecture



Roadway imagery

GPS-tagged frames



Local VLM



Spatial outputs

KML · SHP



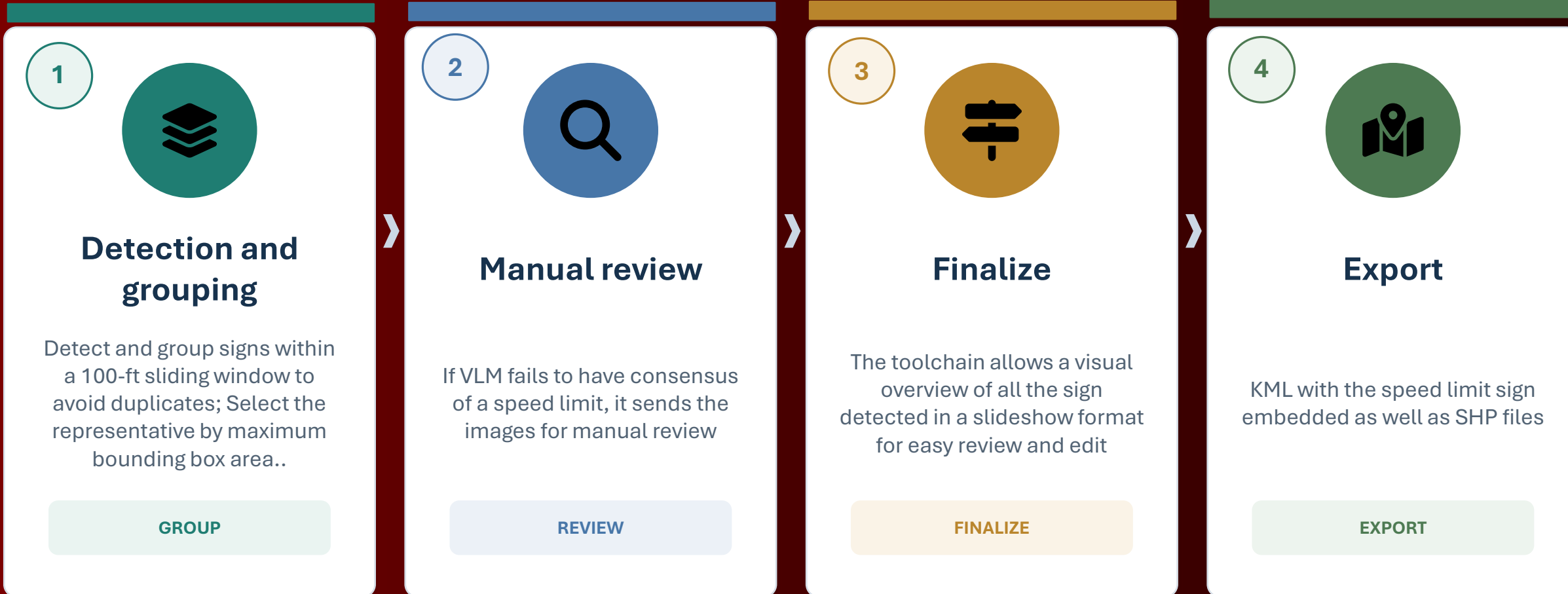
No cloud transfer — proprietary images remained on-device

Creating Posted Speed Limit (PSL) Layer

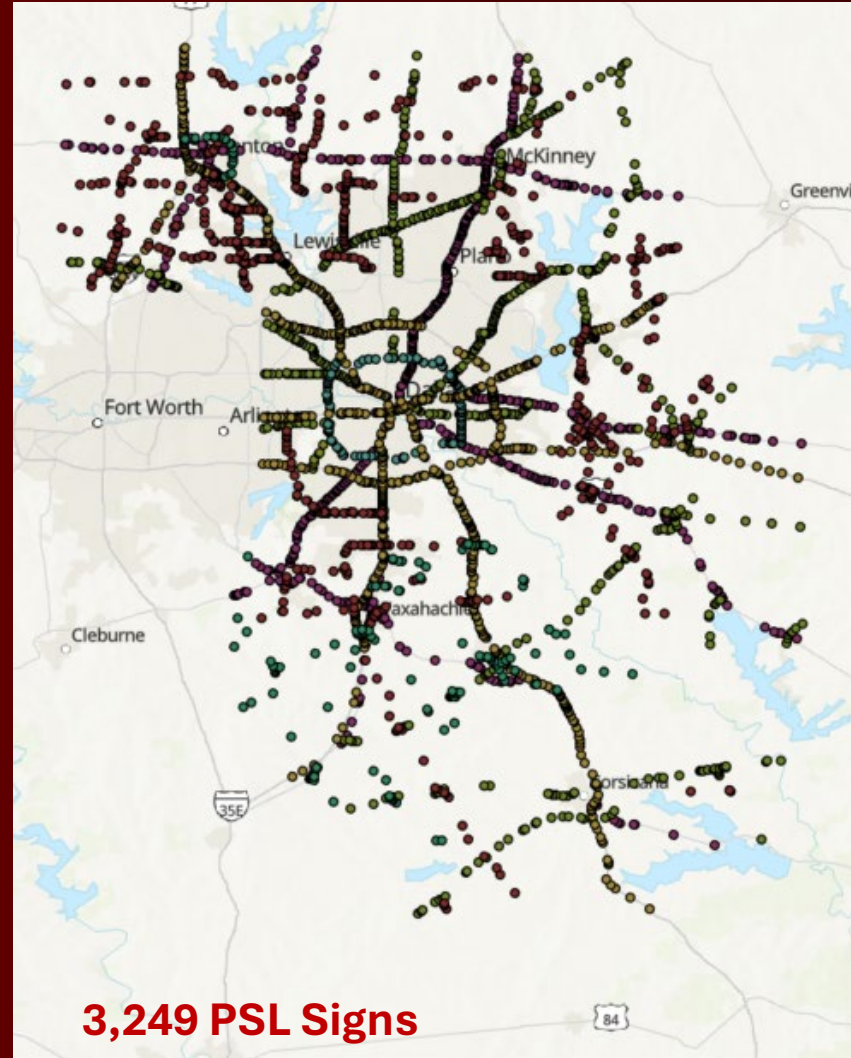
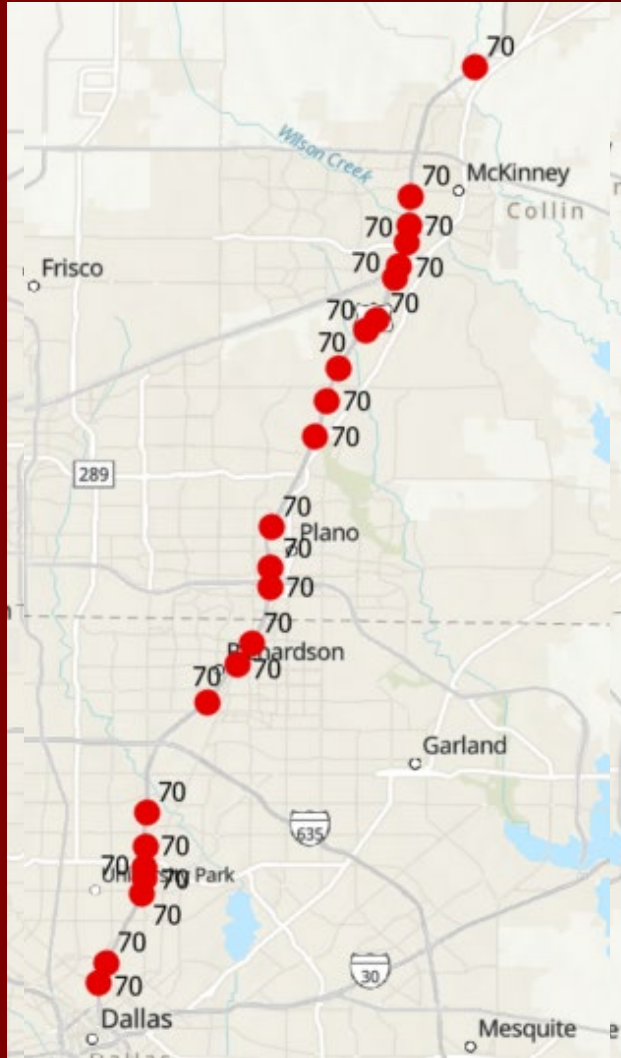
Four-phase workflow



Local VLM



Creating Posted Speed Limit (PSL) Layer



- ~380,000 images (6T data) processed across the TxDOT Dallas District to extract 3,249 PSL signs;
- Total processing time \approx 630–840 hours (\approx 26–35 days) at 6–8 seconds per image (including inference + referencing);
- Processing conducted on a MacBook Pro (M4 Max, 128GB RAM, 14-core CPU, 20-core GPU);
- Full-resolution images used to preserve detection accuracy;
- Further acceleration possible via parallel processing / cluster computing (reduce processing time to a couple of weeks or even days);

TxDOT DAL Speed Zone Study Inventory (2014–2024)

742

Speed Zone Segments

1396

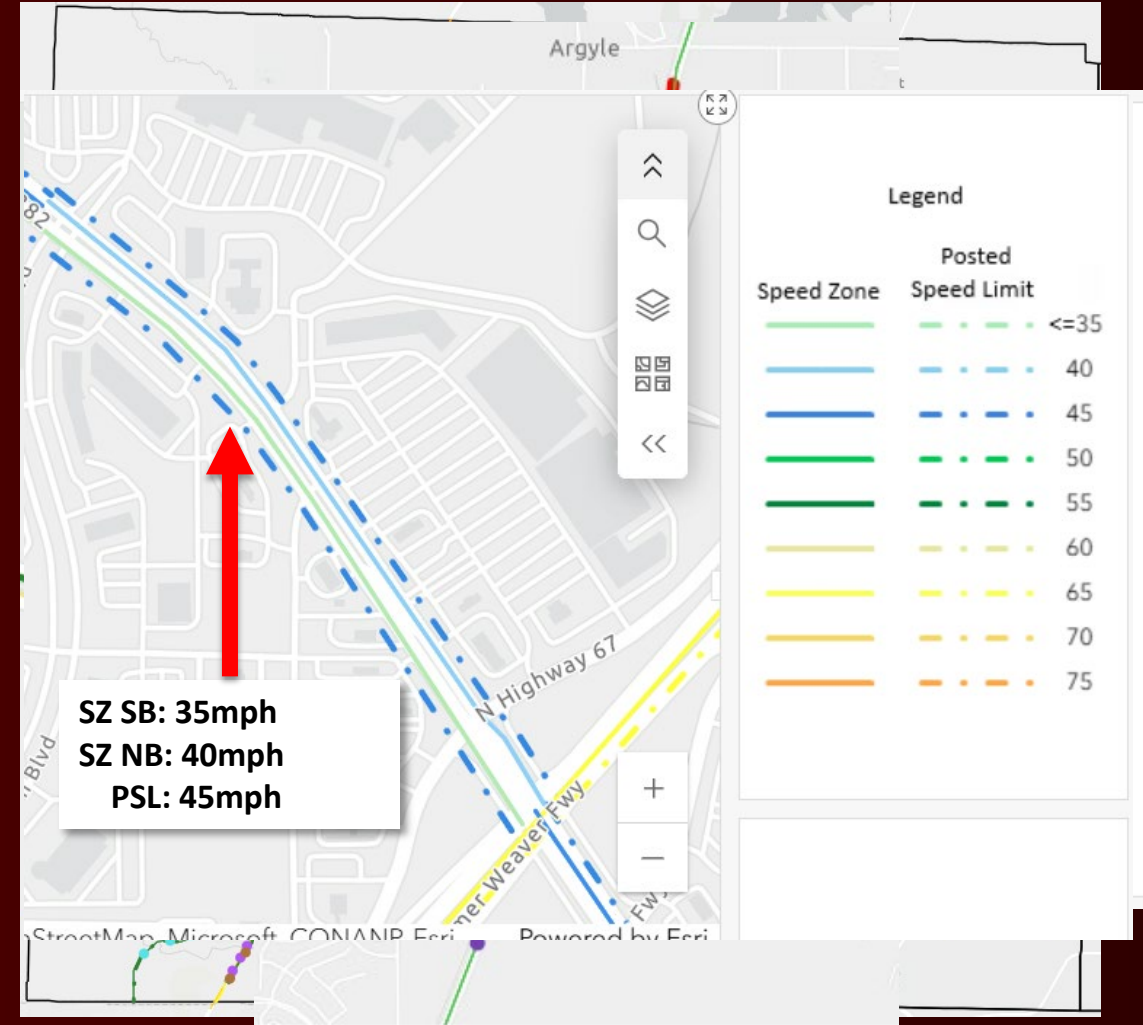
Directional Speed Zone Stations

158

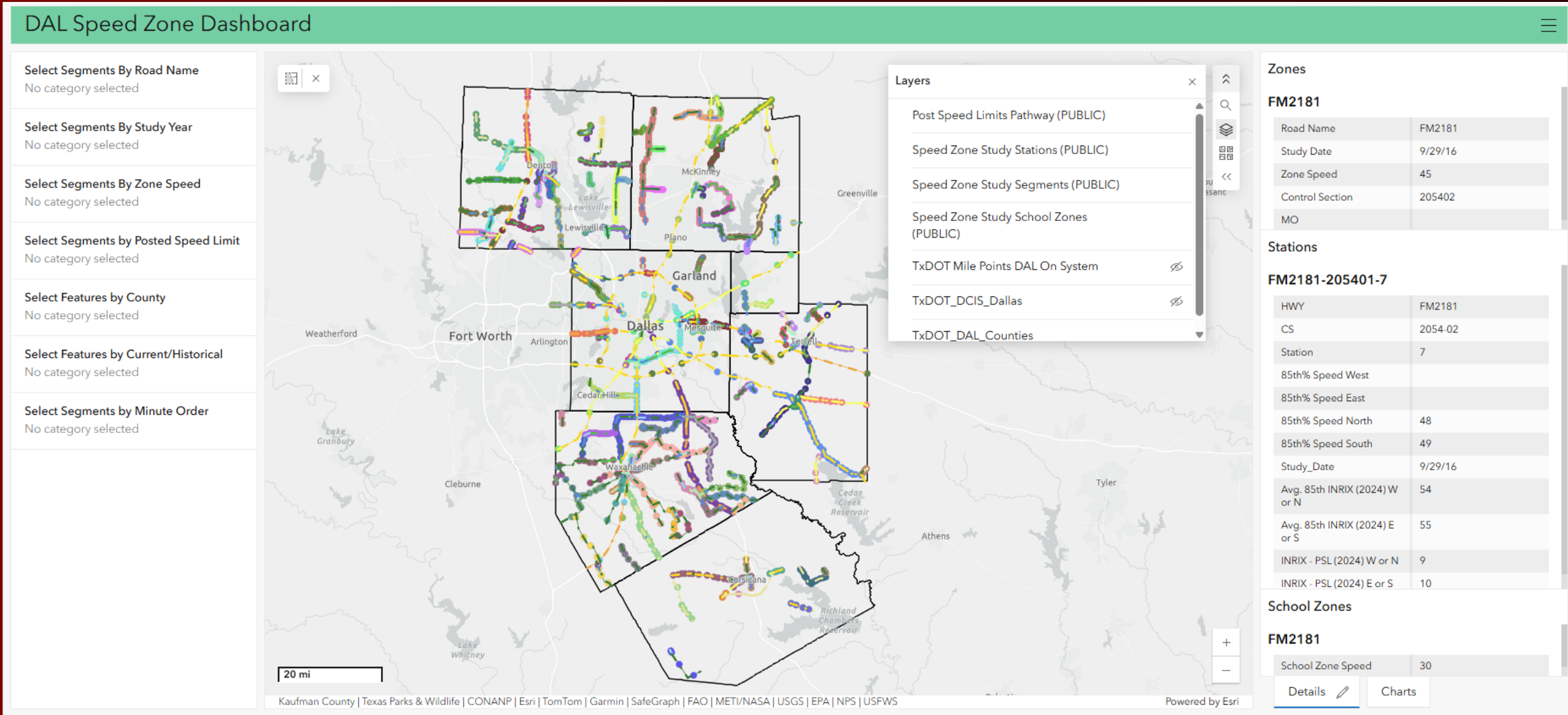
School Speed Zone Segments

589

Posted Speed Limit Segments



TxDOT Dallas District Speed Zone Study Dashboard



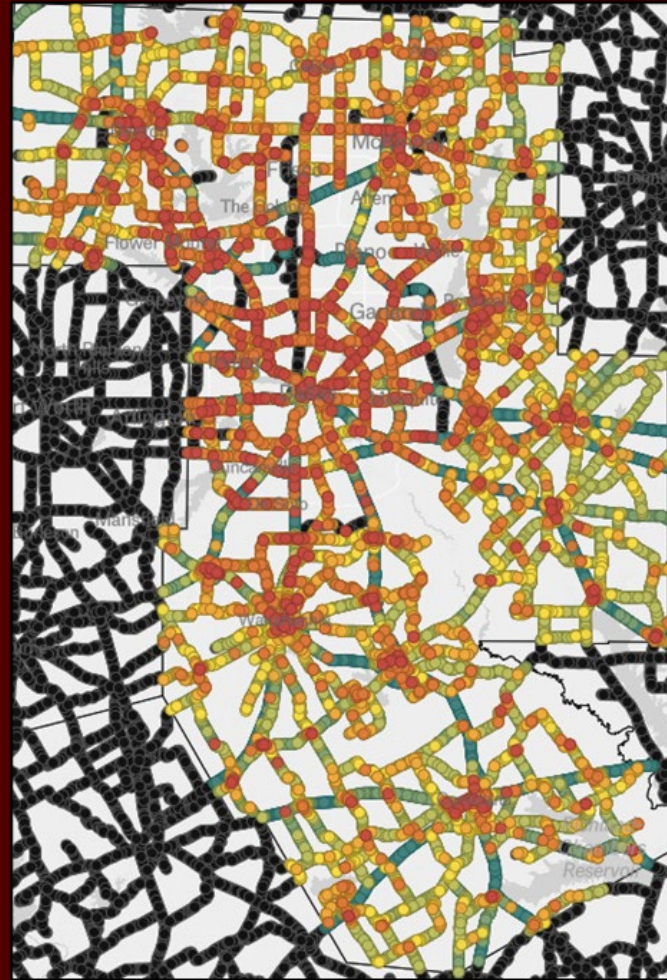
Map Layers

Current

- Speed Zones
- Speed Zone Stations
- School Speed Zones
- Posted Speed Limits (GE/Pathway imagery)
- Mile Point Layer
- 0-7156/Probe Data SSL Layer (Denton)

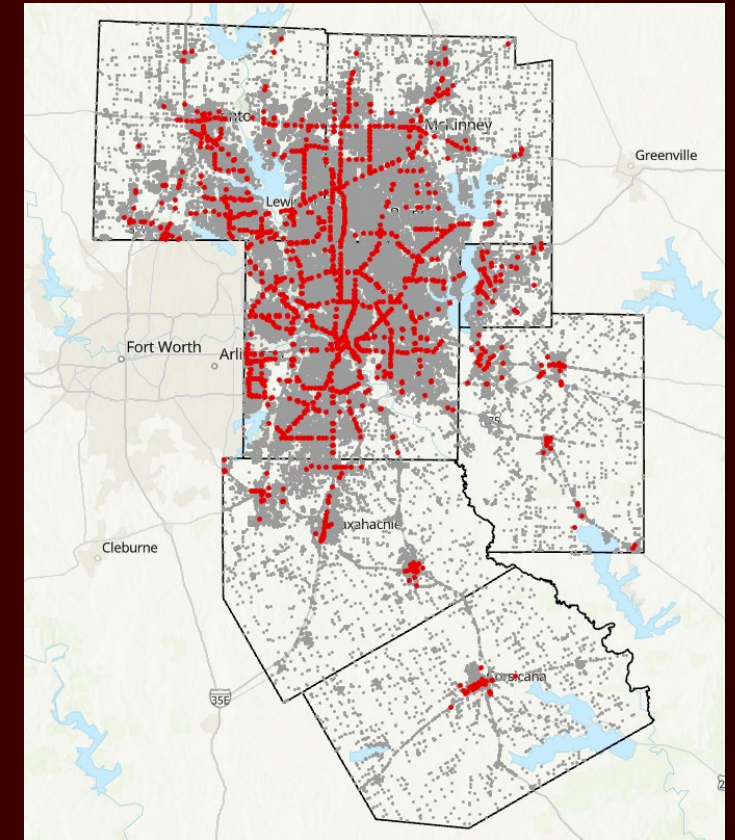
Future

- Wejo Speed Layer
- Intersection Layer
- RIVAL Layer



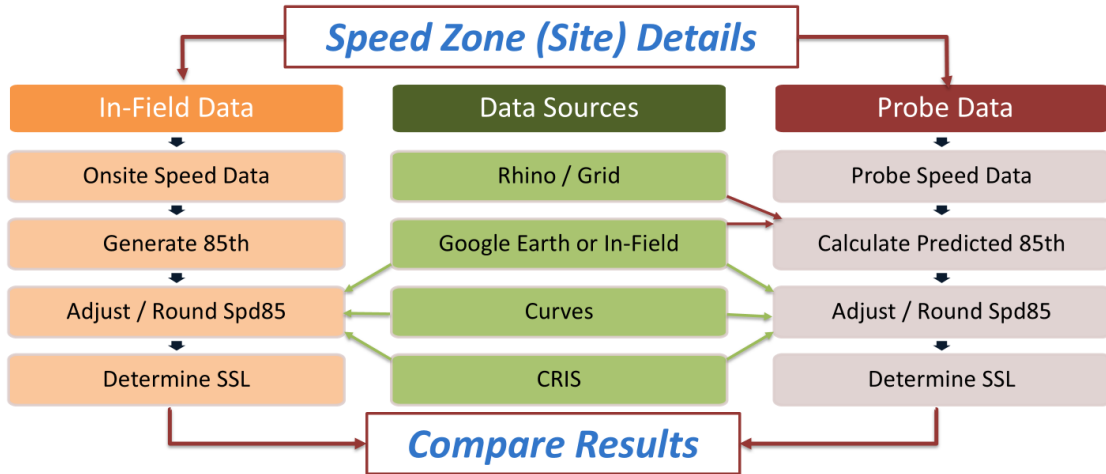
Wejo Speed Layer

Intersection Layer

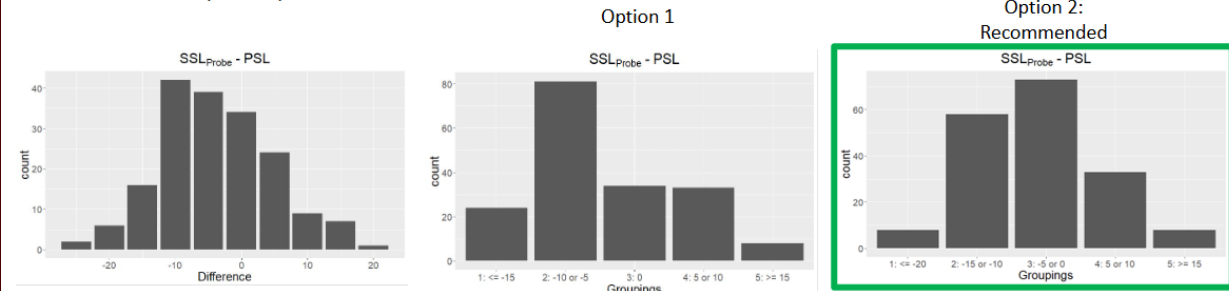


TxDOT Project 0-7156 Integration - Denton County

Suggested Speed Limit Flowchart (Field and Probe)



- Compared
 - Predicted 85th INRIX to Observed Speeds (Stations)
 - $SSL_{(Probe)}$ to PSL (Segments)



0-7156: Using Vehicle Probe Data to Evaluate Speed Limits on Texas Highways

Benefits of the Speed Zone Dashboard



Comprehensive District View

A unified visual environment covering all speed zones, stations, PSLs, and school zones across the entire Dallas District — no more siloed files or disconnected spreadsheets.



Discrepancy Identification

Side-by-side layers make it easy to identify mismatches between zone speeds, posted limits, and statewide planning data — supporting faster corrections and better compliance.



Accessible to AO & Maintenance Staff

Area Office and maintenance personnel gain direct visual access without needing specialized GIS training — streamlining communication and reducing request turnaround time.



Historical Record & Future Planning

Past and present speed study data are stored and indexed, enabling quick review of when corridors were last studied — directly informing budget needs, study scheduling, and records management.



Questions?

Jason Wu, Ph.D.
Minh Le, P.E., PMP
TTI Research & Implementation
Dallas Program



Amanda Miller, P.E.
TxDOT Dallas District
Director of Transportation
Operations

