Using CNN and Spatial-Temporal Embedding for Predicting Smoke PM2.5

Frank Zhao[1], Chenlin Meng, Stefano Ermon, Marshall Burke, David Lobell, Marissa Childs, Jessica Li

Department of Computer Science, Stanford University

Stanford
Computer Science

Research Overview

- Wildfire smoke contributes to **40%** of PM2.5 pollution. Exposure to extreme smoke PM2.5 has increased **27-fold** over the last decade.
- Accurate measurement of smoke-induced PM2.5 is key for understanding the **societal impacts** of wildfire risk.
- Original research uses XGBoost on a location's 42 features to predict smoke PM2.5.
- Instead of only using the location of interest, we look at its **surrounding areas**. Attempt to use CNN to incorporate **spatial information**.
- Based on **Tobler's First Law of Geography**:

"Everything is related to everything else, but near things are more related than distant things."

- Uses location-time embedding to provide **geographical priors** for the model.
- **Problem Statement:** Given a location and its surrounding's features, can we use deep learning to predict its smoke PM2.5 level?
- **Main Contributions:**
  - -- Confirms that **CNNs can incorporate useful spatial information** for atmospheric predictions.
  - -- Confirms that location-time embedding provides a **powerful spatial-temporal prior**.
  - -- Achieves **better results** (increase of 0.04 in R squared metric) with **less features**.

Datasets & Metrics

- Datasets from Marshall Burke's group.
- **400,000 wildfire instances** with the target smoke PM2.5 value. **5 spatial folds.** Folds 1-4 for training. Random half of fold 0 for validation / testing.
- 10 features with too many missing values are **discarded**.
- Each instance **processed into 11x11x32 "images"** to include spatial information.
- **Smoke PM2.5** defined as PM2.5 pollution above the monthly median on a smoke day.
- Overall objective is to minimize the **Huber Loss**, which is more robust to outliers.

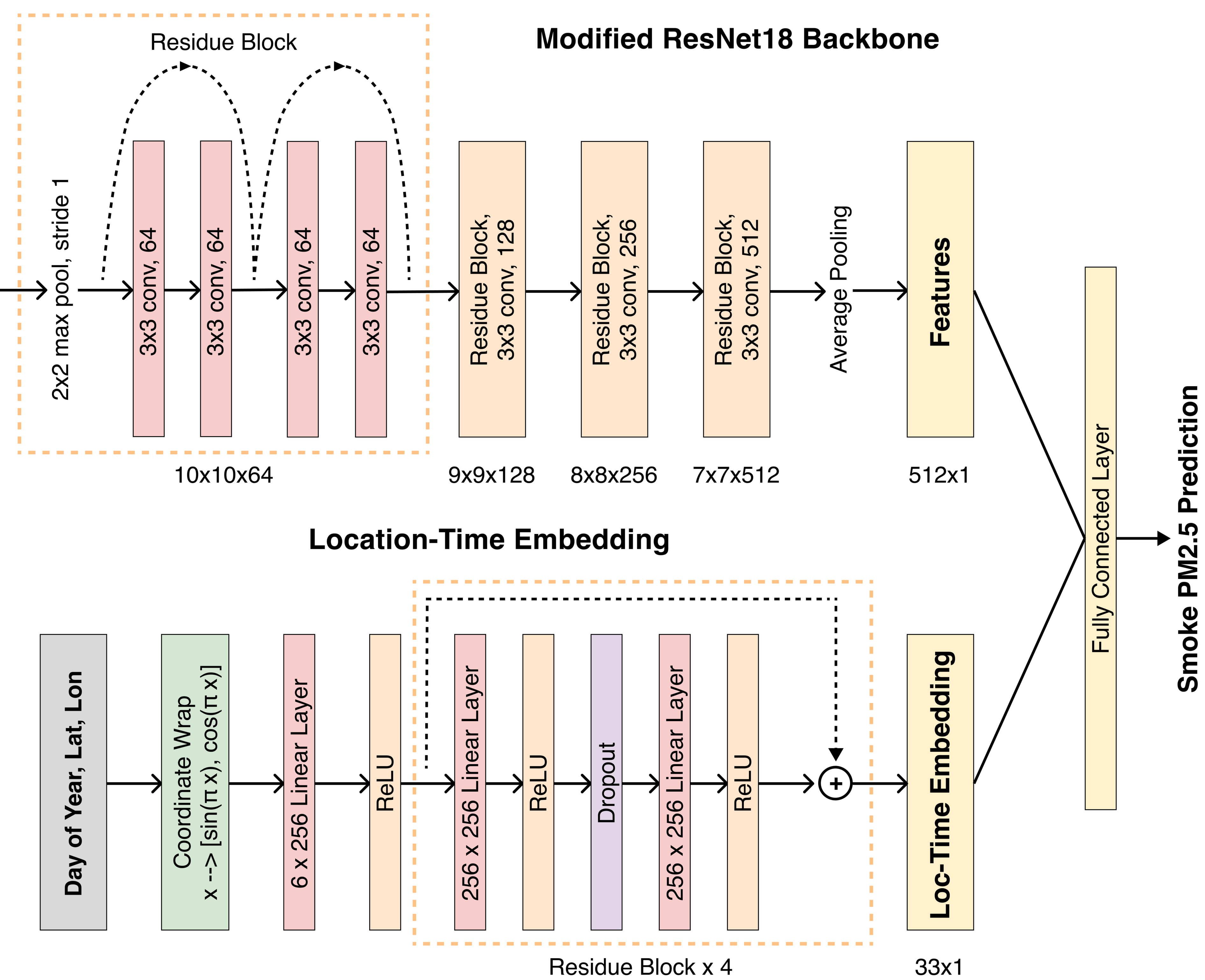
$$l_n = \begin{cases} 0.5(x_n - y_n)^2, & \text{if } |x_n - y_n| < delta \\ delta * (|x_n - y_n| - 0.5 * delta), & \text{otherwise} \end{cases}$$

Methods & Model Architecture

10 km x 10 km grids, each with 32 features:

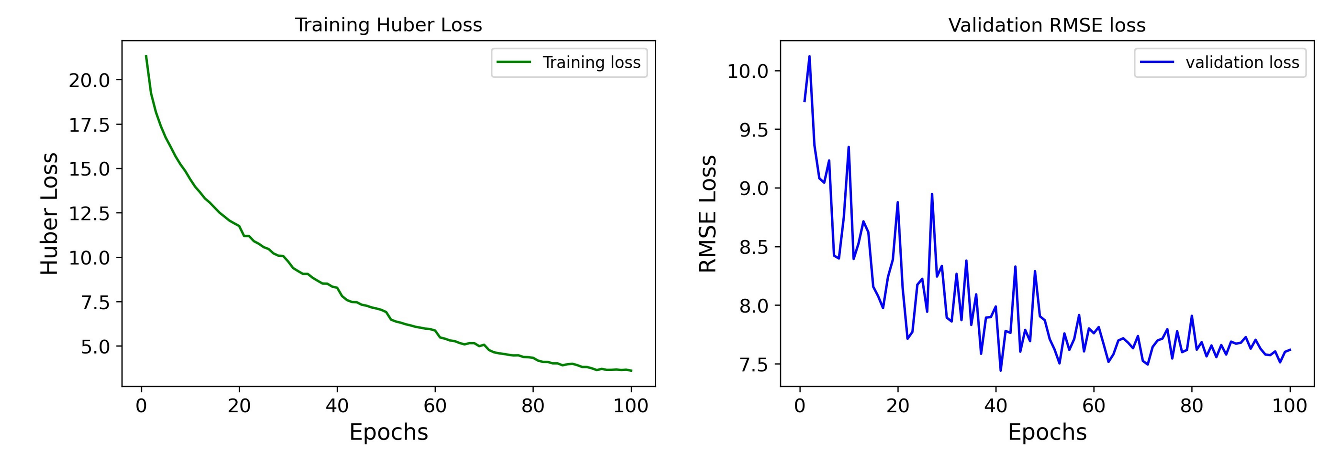
- Geographical Info
- AOT Anomalies
- Fire Variables
- Meteorology

The modified ResNet18 reduces the width and height by -1 instead of /2 each time.



- Coordinate wrap makes Dec. 31st close to Jan. 1st.
- Target variable split into 33 bins for classification.
- Embedding network pre-trained for 100 epochs.
- Optimize for presence-only cross entropy loss.
- It is allowed to update during actual training.

Results

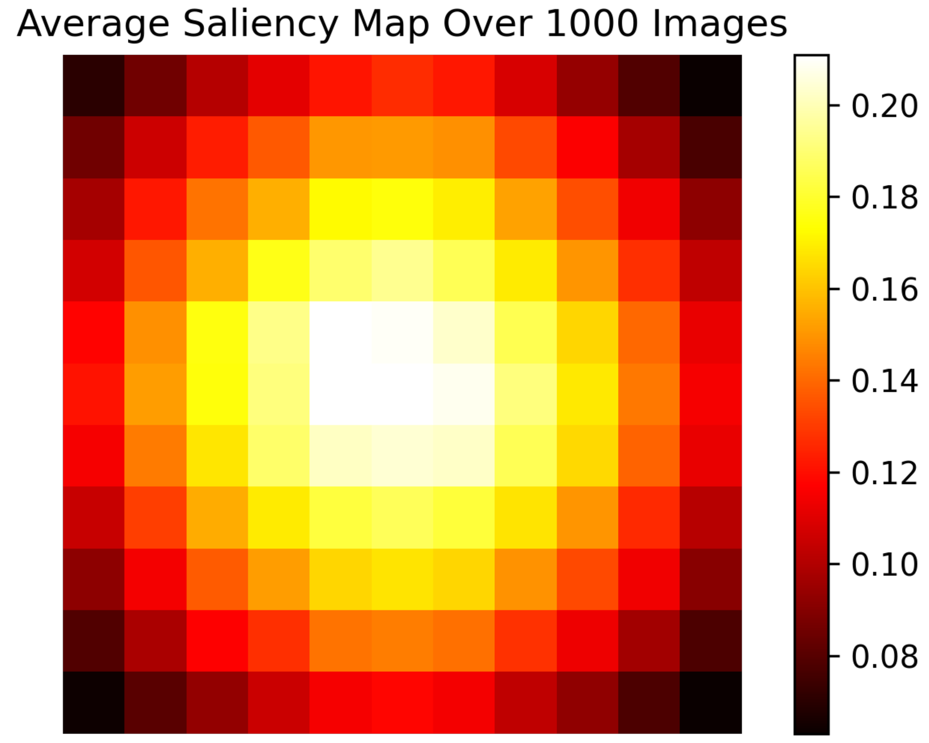


	RMSE Loss	R Squared
Partial XGBoost (32 features)	8.697	0.602
Full XGBoost (42 features)	8.185	0.629
Original ResNet18	8.921	0.581
Modified ResNet18	8.554	0.615
+ Loc-Time Embedding	8.534	0.617
+ Data Augmentation	8.147	0.641
Change RMSE to Huber Loss	**7.880**	**0.664**

- Increase from adding Loc-Time Embedding may seem minimal here but it performs much better on val set.
- Data Augmentation includes horizontal and vertical flips since they preserve spatial information.
- Best architecture uses **Modified ResNet18 + Variable Location-Time Embedding + Data Augmentation + Huber Loss.**

Evaluations & Future Work

Saliency Map



Confirms Tobler's First Law of Geography!!

Feature Importance

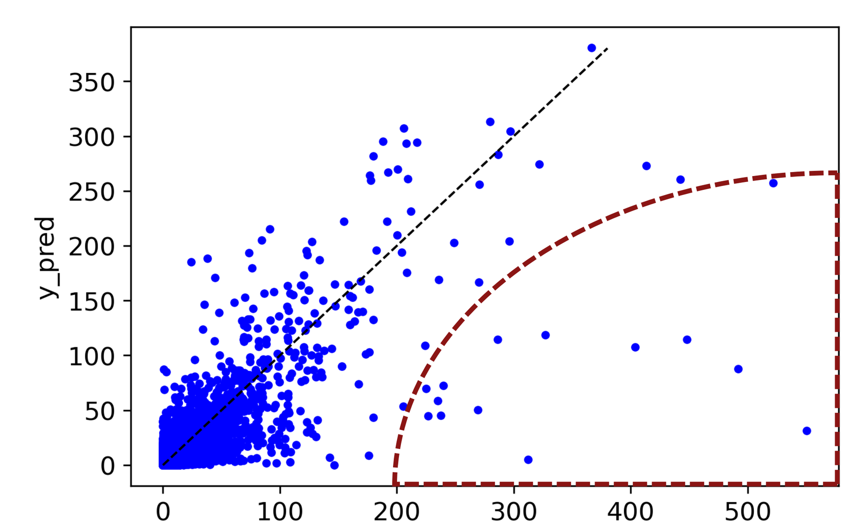
Feature Type	Mean Attribution
"Image" Features	0.243
Loc-Time Embedding	0.142

- Integrated gradients method in Captum.
- Location-time embedding is quite useful!

Some of the most important features include:

- AOT anomalies
- Dewpoint temperature
- Month, Lat, Lon
- Distance to closest fire

Imbalance in Target



- The target is highly skewed. 95% in [0, 20] and 5% in [20, 800].
- R squared for y < 200 is 0.630 compared to -2.56 for y > 200.

Failure Case

- Target is 312, but prediction is 5.18.
- September, near Yellowstone.
- Medium-size fire.
- Probably due to Low AOT anomalies on the day and on previous days.
- A spike in PM2.5 pollution that day.

Future Work

- Results can be variable due to imbalance. Average over random splits to confirm effectiveness of each modification.
- Address imbalance by finding new features that can distinguish edge cases.