DEEP LEARNING FOR DRIVING DETECTION ON MOBILE PHONES

Chetan Ramiah, Allen Tran, Evan Cox and George Mohler

August 14, 2016

Metromile

Usage Based Insurance

Mileage is a large component of automotive risk yet almost all premiums are a flat fee per time period

- Low mileage drivers subsidize high mileage drivers

Usage based insurance charges in proportion to mileage.

Metromile

- Pulse device (GPS + other sensors) installed in OBDII port in vehicle

Introduction

Mobile phones have an array of sensors that enable smart driving apps

- usage-based insurance
- safe driving

This paper: detect driving activity in an energy-efficient manner using Convolutional Networks

Findings

1. In activity classification, deep learning beats traditional ML approaches after $\approx$ 500 hours of data
2. Conv. nets can distinguish between modes of automotive behavior (0.8 AUC)
3. Conv. net to detect driving implemented on phone (8MB storage, 4 seconds at test time)

Existing approaches

Feature generation + standard classifiers

- Trees and ensembles of trees, SVM, KNN, Naive Bayes, MLPs

Typical features

- sliding windows, moments of raw time series/FFT
- auto-regressive coefficients

Key advantage of Deep Learning approach here: no need to do feature engineering

Model Overview

Classification via a convolutional network on spectrograms of sensor readings

1. activity classification (still/walking/automotive)
2. driving classification within automotive (driving/not driving)

Challenges:

- has to be energy efficient (low power sensors vs GPS)
- sensor hardware/quality differs across phones
- sensor readings are oriented relative to the device

Data

Random two minute samples from 2,133 drivers of sensor readings using Android consumer smartphones

5,400 hours of accelerometer and gyroscope readings

- removed sleeping/at rest readings
- interpolated at 50hz for acc and 10hz for gyro to neutralize hardware differences
- split into 30 second windows w/ 50% overlap

Experiments

Two supervised learning experiments

1. Activity Classification (walk or still or automotive)
   - labels from Android platform
2. Driving detection (driving or not)
   - labels derived from iBeacon device installed in vehicle

Experiments: Benchmark

Model based on Hemminki (2013) used as benchmark

- Random Forest with hand engineered features

Table: Baseline features

Domain	Features
Statistical	Mean, STD, Variance, Median, Min, Max, Range, Interquartile range Kurtosis, Skewness, RMS
Time	Integral, Zero-Crossing Rate
Frequency	FFT DC first five frequency responses, Wavelet Entropy, Wavelet Magnitude

Experiments: models

Models based on convolutional networks. Why?

- learn features from raw input
- parameter sharing reduces free parameters
- raw sensor input has notion of “local” regions (time, frequency)

CNN Architectures

1. spectrogram or raw sensor time series as input
2. multi-stream

Single stream CNN

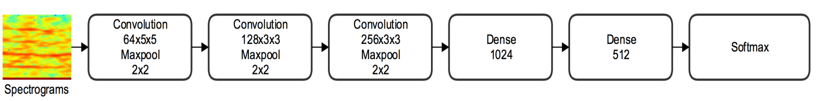

Figure: spectrogram

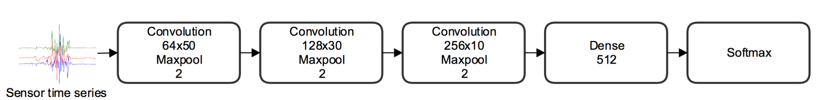

Figure: time series

Multi-Stream CNN

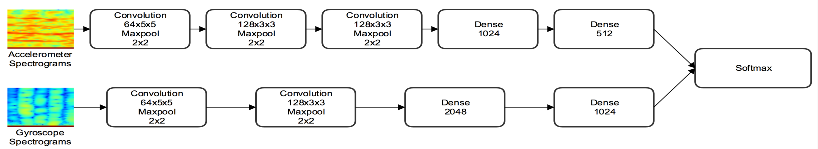

Figure: multi-stream spectrogram

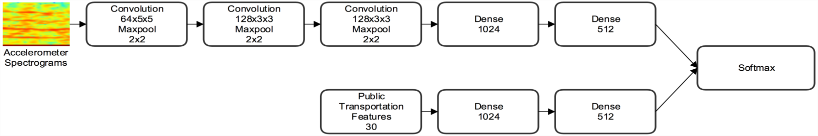

Figure: multi-stream spectrogram

Spectrograms

Spectrograms, from rolling FFTs, are useful representations of sensor input

- time on horizontal, frequency on vertical axis
- color/value represents amplitude at that frequency and time window

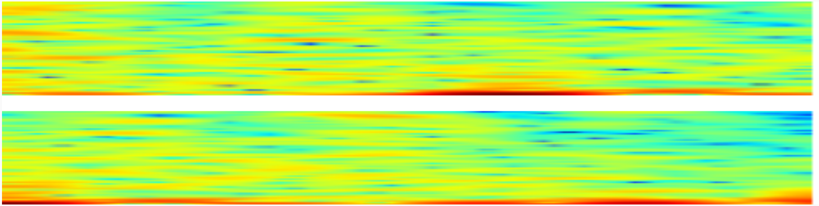

Figure: multi-stream spectrogram

Spectrograms are good initial representations

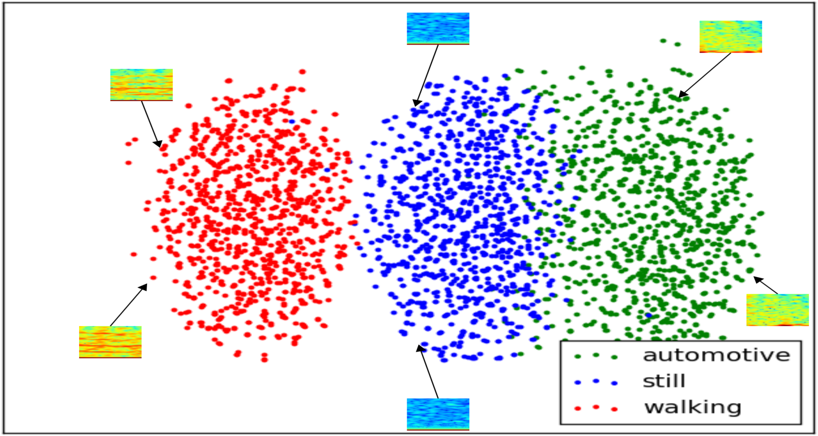

Figure: T-SNE visualization of the activity recognition problem.

Activity Classification

Table: Activity recognition results

Approach	AUC
Baseline	0.84
Accelerometer Spectrogram	0.89
Accelerometer Time Series	0.83
Accelerometer and Gyroscope	0.89
Accelerometer and GPS	**0.91**

Activity Classification: Confusion Matrix

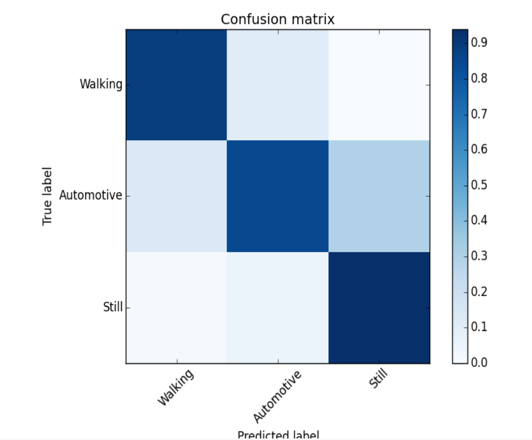

Figure: Confusion matrix for activity recognition

Activity Classification: size of data

Deep Learning wins $\geq$ 500 hours

Table: AUC scores for activity recognition with varying dataset sizes

Dataset Size	Baseline	Accelerometer CNN
50 hours	**0.82**	0.78
150 hours	**0.82**	0.82
500 hours	0.84	**0.86**
1000 hours	0.84	**0.88**
1300 hours	0.83	**0.89**

Driving detection

Subset of data labeled automotive and

- driving labels from iBeacon device installed in vehicles
- restricted to single vehicle policies

362 unique vehicle types

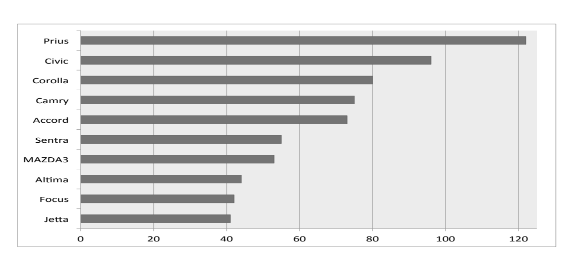

Figure: Frequency of top ten vehicles in driving detection dataset.

Table: Driving detection results

Approach	AUC
Baseline	0.67
Accelerometer Spectrogram CNN	0.77
Accelerometer Temporal CNN	0.75
Accelerometer and Gyroscope CNN	0.77
Accelerometer and GPS CNN	**0.80**

Conclusion

Sensor input $\rightarrow$ Spectrogram $\rightarrow$ Convolutional Network works well for driving detection