# Distracted Driver Kaggle Competition

Chen Chen

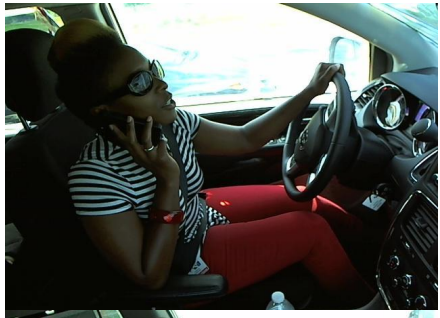
Nelson Chen

Nesty Torres

Arash A. Omrani

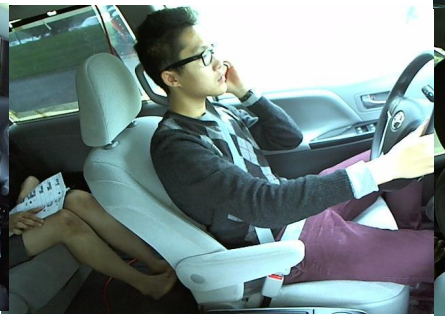
Mentor: Adam Kalman












# Outline

- Problem statement
- Evaluation and Kaggle leaderboard
- Approaches
    1. Logistic Regression and Neural Network
    2. Convolutional Neural Network
    3. Convolutional Neural Network with pre-trained initialization
- Summary of results

# Problem Statement

- CDC data – 1 in 5 car accidents is caused by a distracted driver
- Can computer vision spot distracted driver?
- Given - dataset of 2D dashboard camera images
- Can we classify each driver's behavior? (10 categories)

* Goal – predict the likelihood of what the driver is doing in each picture *

# Evaluation and Kaggle leaderboard

$$logloss = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{M} y_{ij}\log(p_{ij})$$

$N$ ---- Number of test images

$M$ ---- Number of class labels

$y_{ij}$ ---- 1 if observation $i$ belongs to class $j$, 0 otherwise

$p_{ij}$ ---- predicted probability that observation $i$ belongs to class $j$

# Evaluation and Kaggle leaderboard

$$logloss = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{M} y_{ij}\log(p_{ij})$$

$N$ ---- Number of test images

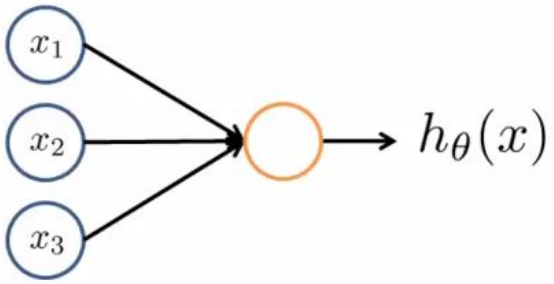
$M$ ---- Number of class labels

$y_{ij}$ ---- 1 if observation $i$ belongs to class $j$, 0 otherwise

$p_{ij}$ ---- predicted probability that observation $i$ belongs to class $j$

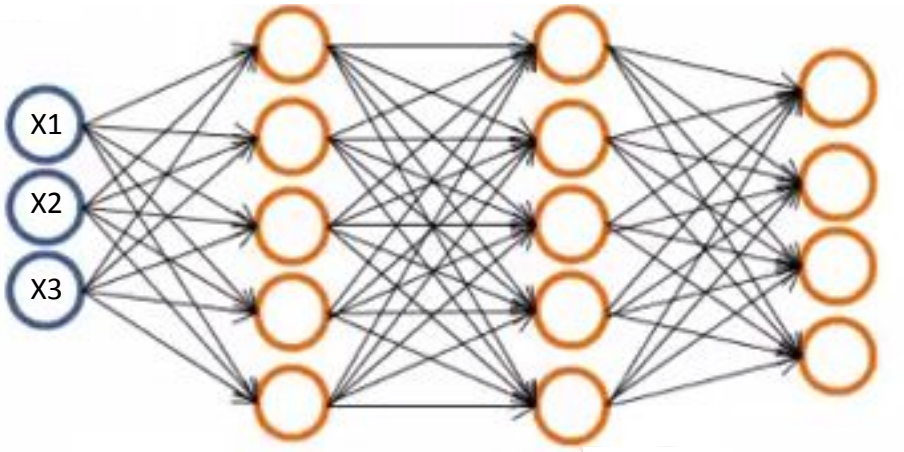
| # | Δ1w | Team Name * in the money | Score | Entries |
|---|---|---|---|---|
| 1 | ↑1 | Z_B_C * | 0.09275 | 101 |
| 2 | ↓1 | MakeAmericaGreatAgain * | 0.09488 | 179 |
| 3 | ↑2 | BR BRAZIL POWER BR * | 0.10208 | 178 |
| ⋮ | | | | |
| 352 | ↓37 | OP | 0.60589 | 7 |
| **353** | **↑32** | **CDIPS** | **0.60691** | **33** |
| 354 | ↓38 | Meepo | 0.61160 | 5 |
| ⋮ | | | | |
| 1448 | ↓91 | Silvio | 30.35184 | 2 |
| 1449 | ↓91 | Chris Jepeway | 30.74658 | 1 |
| 1450 | ↓91 | Fernando | 30.94094 | 1 |

# Logistic Regression and Neural Network




Input Output

Logistic Regression

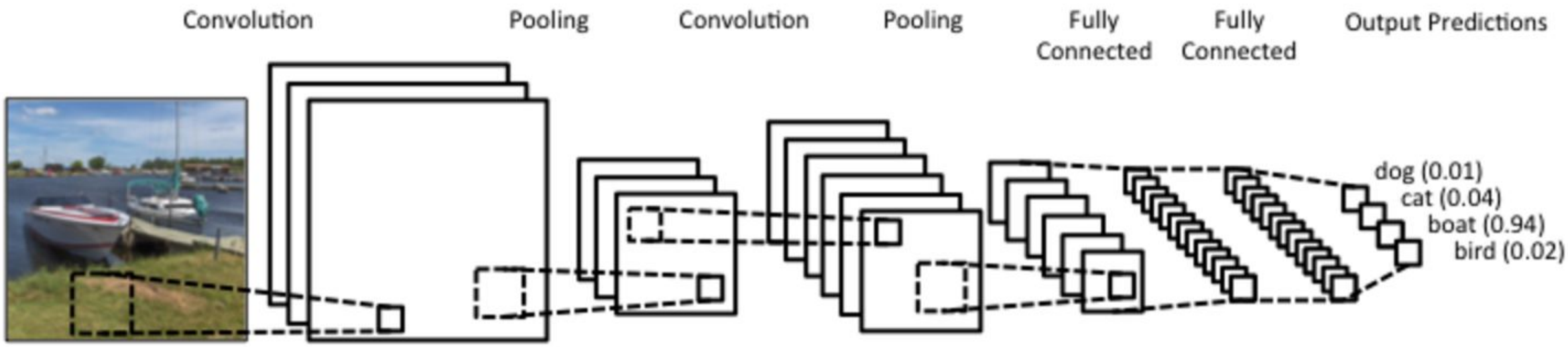



Input Hidden layer 1 Hidden layer 2 Output

Multi-class Neural Network

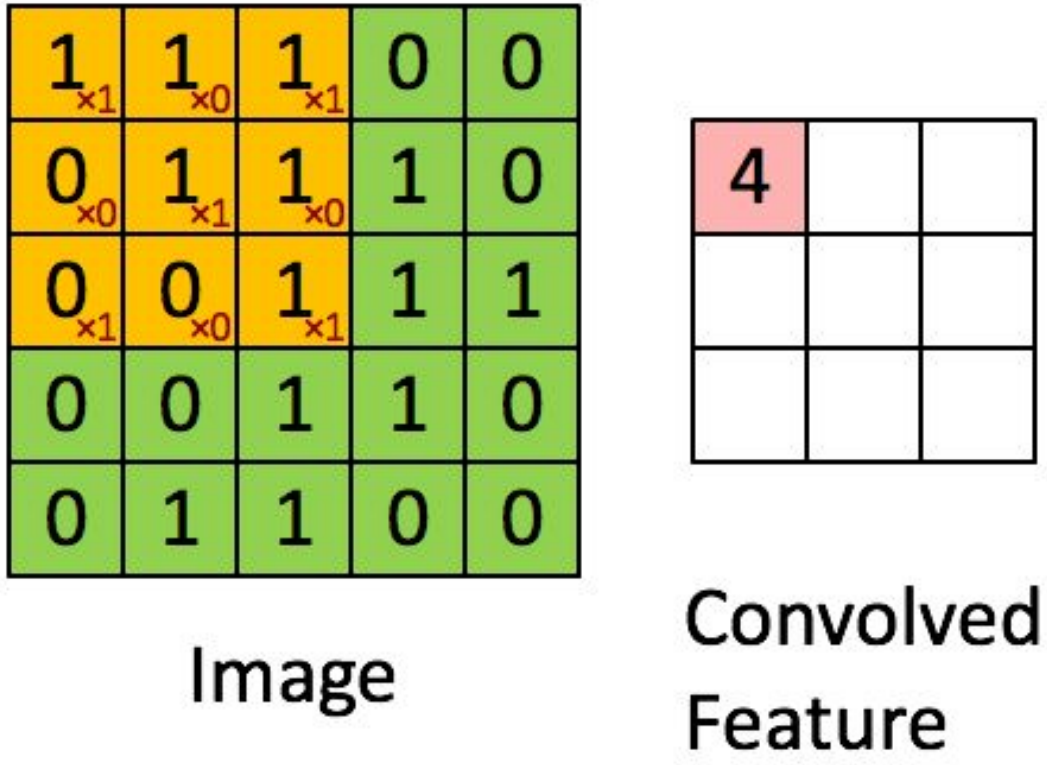
- Best LR score: 4.2
- Best NN score: 2.6, rank: 1004
- NN fits very well on the training set and even on cross-validation. But high error on test set
- There are different drivers in test-set

# Convolutional Neural Networks (CNN)




http://www.wildml.com/2015/11/understanding-convolutional-neural-networks-for-nlp/

# Convolutional Filtering




http://www.wildml.com/2015/11/understanding-convolutional-neural-networks-for-nlp/

# Examples of Filters

Averaging nearest neighbors

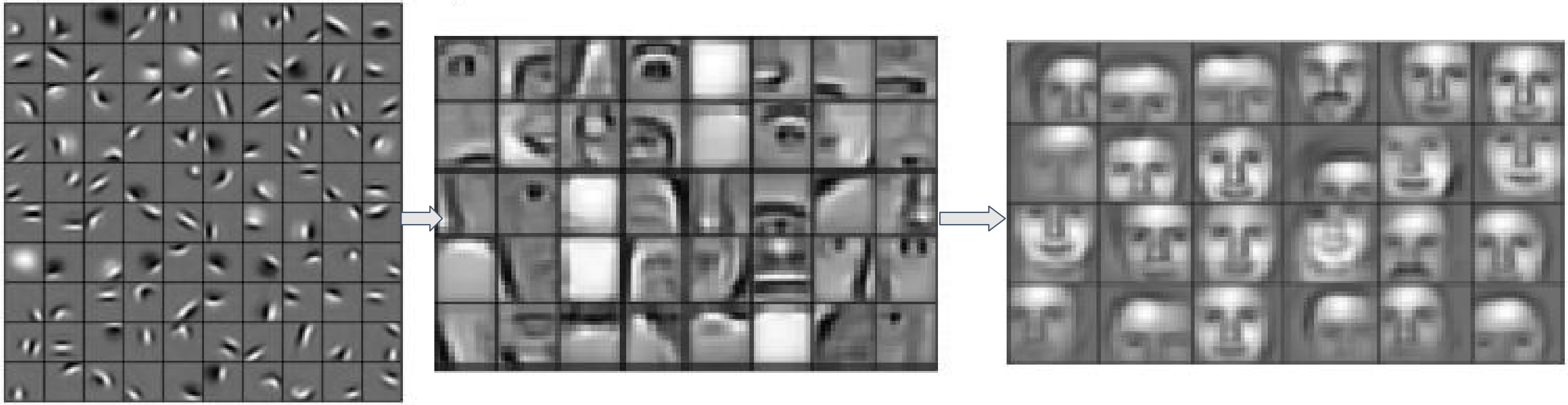
| | | | | |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 1 | 0 |
| 0 | 1 | 1 | 1 | 0 |
| 0 | 1 | 1 | 1 | 0 |
| 0 | 0 | 0 | 0 | 0 |

Difference from neighboring pixels

| | | | | |
|---|---|---|---|---|
| | | | | |
| | 0 | 1 | 0 | |
| | 1 | -4 | 1 | |
| | 0 | 1 | 0 | |
| | | | | |

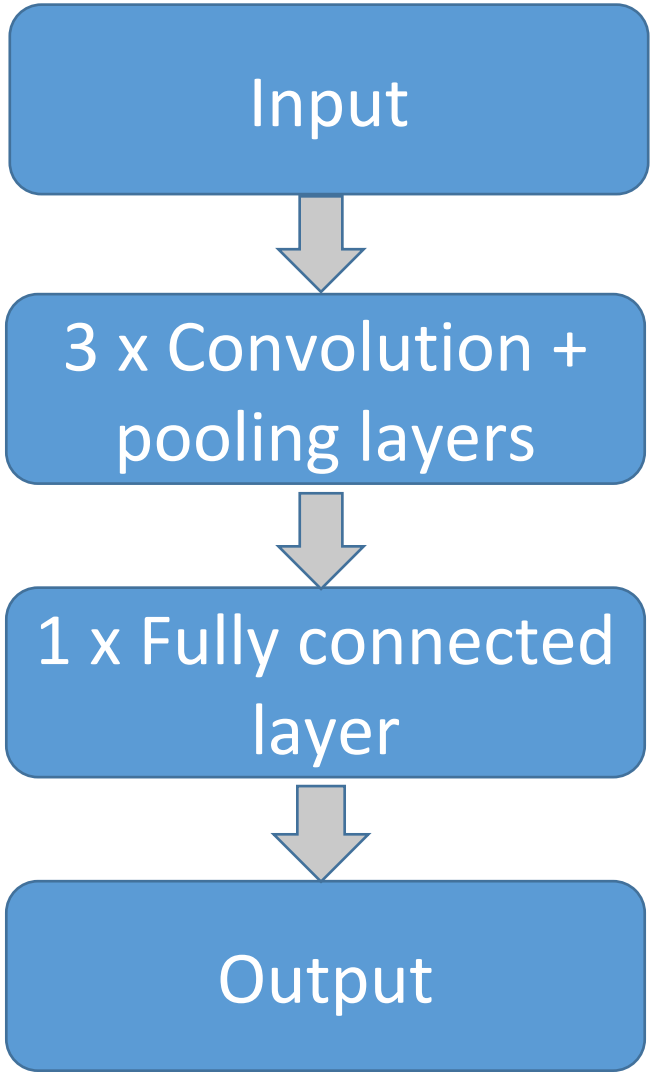
http://www.wildml.com/2015/11/understanding-convolutional-neural-networks-for-nlp/

# Facial Recognition Example



https://devblogs.nvidia.com/parallelforall/deep-learning-nutshell-core-concepts/

# Custom Built CNN




## Computational details:

- ➔ Resized to 64x64
- ➔ Used Keras/Theano library backend (built on C)
- ➔ Took 4-5 hours to run on my laptop
- ➔ Took 20 mins to run on GPU Amazon Instance
- ➔ Took a little more than twice as long to run for 128x128

# Custom Built CNN Results

Results:

- Score: 0.78
- Leadership board: 406$^{th}$ Place

(500+ ranks up!)

Problems:

- Far from the top score
- Can’t improve anymore because of overfitting

# CNN with Pre-trained Model

- “VGG_16” model: 13 convolutional layers + 3 fully connected layers
- Model trained on millions of images from ImageNet competition
- Load pre-trained weights
- Fine tune the weights on the distracted driver dataset
- Score: 0.61 after only 1 epoch. Our best score!
- Problems: takes too much RAM and time! We used up all our AWS credit!

# Feature Heat maps


https://www.kaggle.com/c/state-farm-distracted-driver-detection/forums/t/21994/heat-map-of-cnn-output

# Future Work

- More training/fine tuning with pre-trained models
- Combine different models to improve the score
- More data augmentation

Thank you!
Drive Safe!