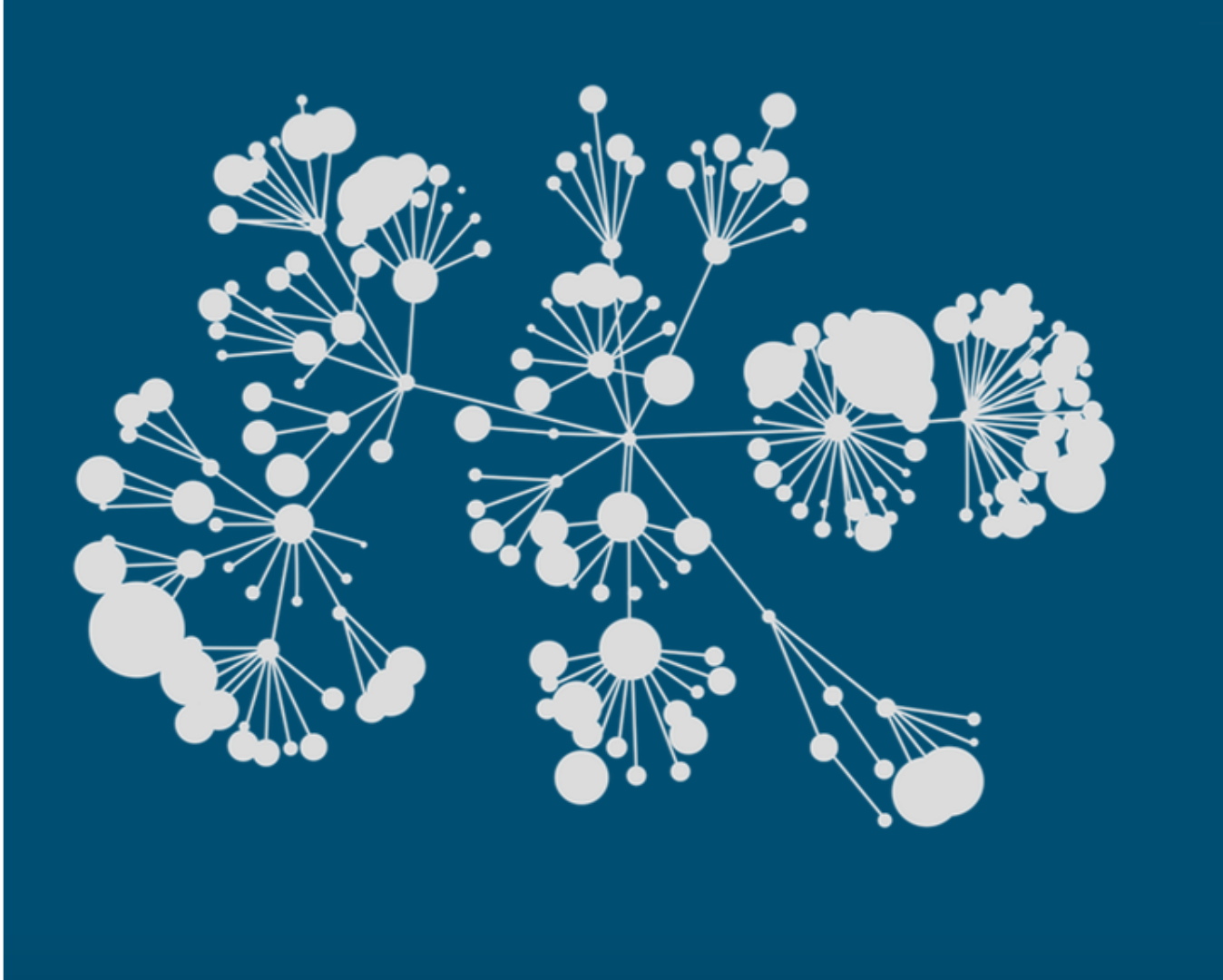
Kaggle

Passenger Screening Competition

Michael Avendi PhD

kaggle



Agenda

1. Background
2. Summary
3. Model Architecture
4. Training methods
5. Important findings
6. Simple model

Summary

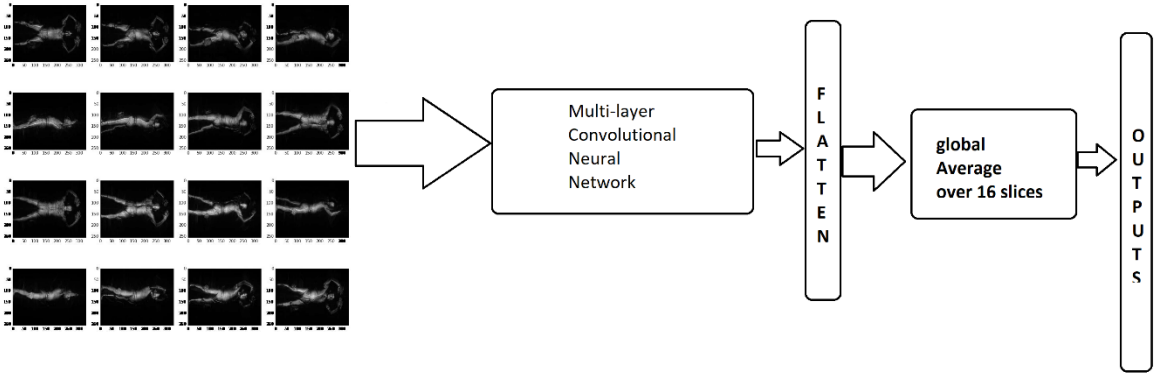
- End-to-end convolutional neural networks.
- Model input: 16 images per subject.
- Model output: 17 probabilities per zone.
- Ensemble of 11 models.
- Only APS data format was used.
- Training on Titan X GPUs.
- Training time: ~48 hours per model.

TOOLS

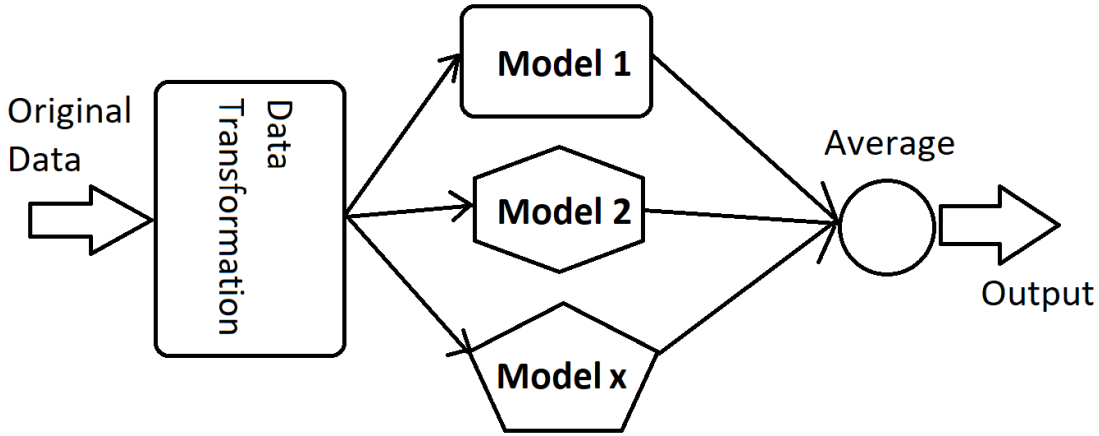
- Anaconda with Python 2.7
- Jupyter notebook
- Numpy: 1.12.1
- Keras 1.1.1
- Theano 0.9.0
- Cv2 3.1.0
- Sklearn 0.18.1

Model Architecture

- Single Model Architecture



- Ensemble Models

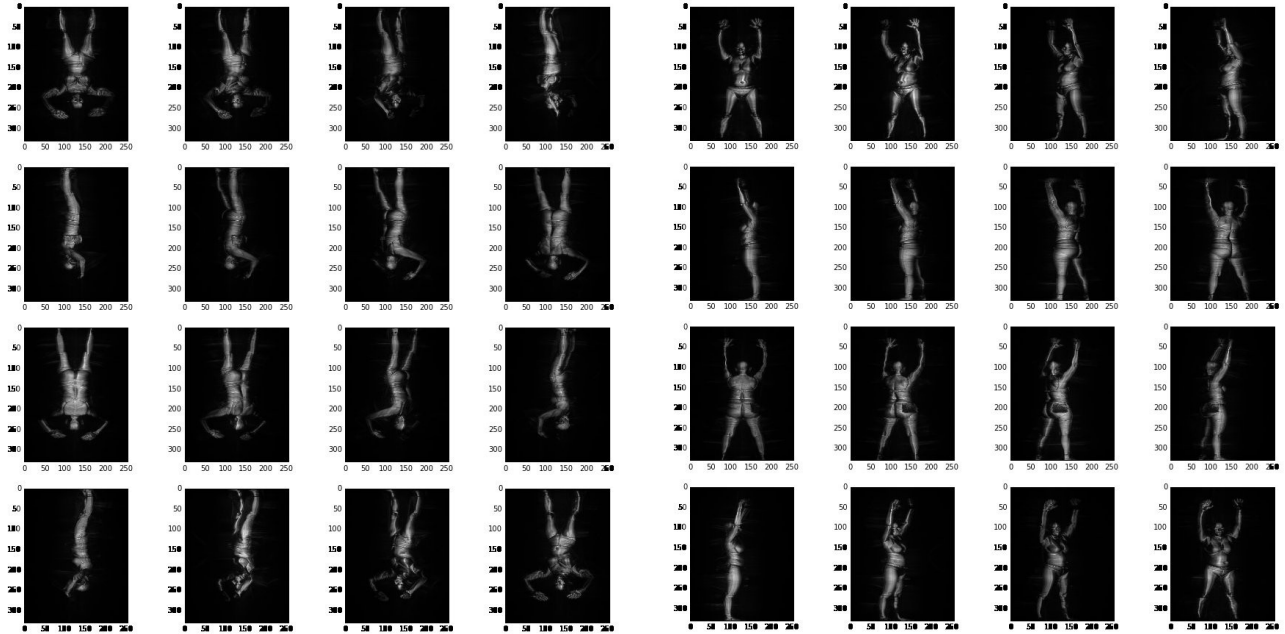


Data transformation

- Convert APS data to Numpy float32 arrays
- Down-sampled to 256 by 330 for nine models
- Original 512 by 660 for two models
- Rotate data by 90, 180 and 270 degrees
- Global zero-mean and unit-variance normalization

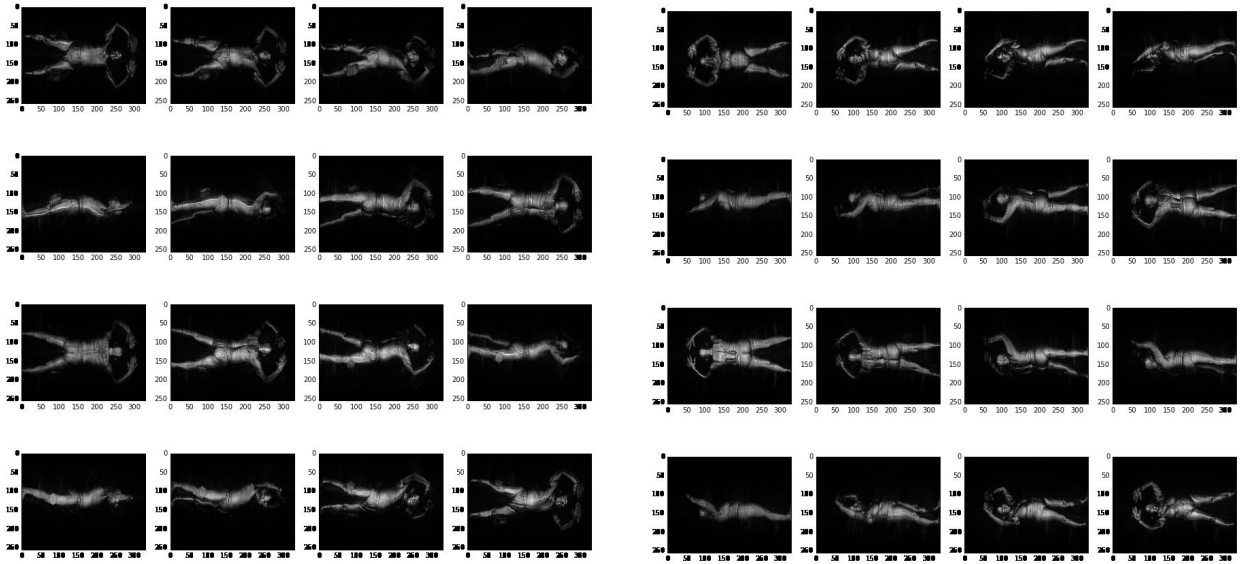
Data Transformation

Rotated data 90 and 270 degrees



Data Transformation

- Rotated data 0 and 180 degrees

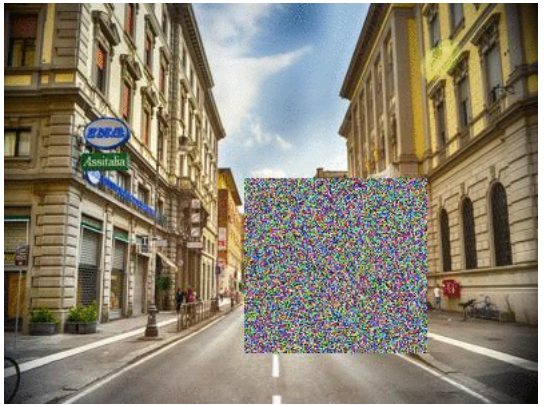


Training Methods

- Stage1-train: 1047 subjects for training and 100 subjects for local validation.
- Stochastic gradient descent algorithm
    - Optimizer: *Nadam*
    - Loss: Binary cross entropy
    - Batch size: 8
    - Epochs: 1000
    - Initial learning rate: 3E-4
    - Drop learning rate by factor of 2 at saturation

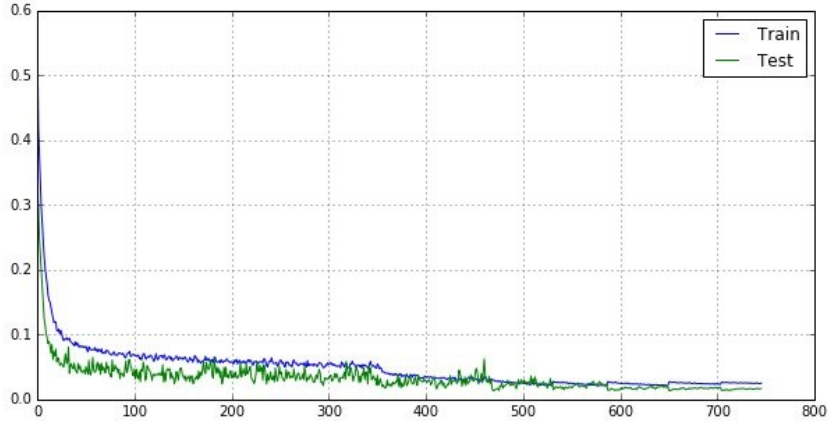
Various image augmentations techniques:

- Rotation (10-15 degrees)
- Width and height shift (10-15%)
- Shear, zoom (10-15 %)
- Random erasing [Ref1]



Models performed almost similarly!

- Single model on stage1-leaderboard ranged from 0.0157 to 0.0297



Insights and findings

- Building separate models on 90,180,270 rotated dataset was very effective.
- Random erasing augmentation provided some improvements.
- Using the original data size.
- Using batch normalization.
- Fine-tuning models after releasing labels for stage1-test data.

kaggle™