# Facial Keypoints Detection

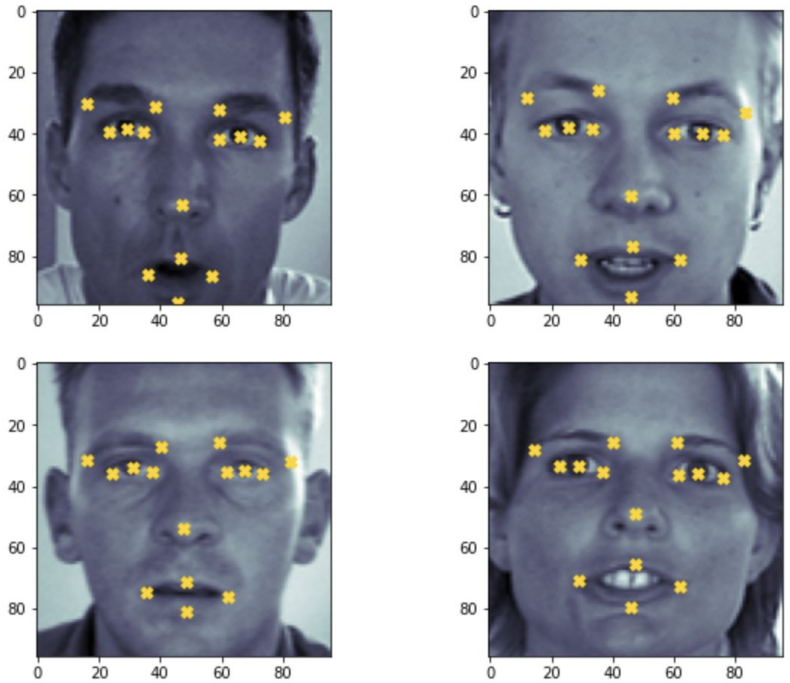
Jordan Meyer
Aswin Thiruvengadam
Sweta Bhattacharya
Leon Rafael Gutierrez Angulo

# Objective:

## Detect the location of keypoints on face images[1]

Applications:

- Tracking faces in images and video
- Analysing facial expressions
- Detecting dysmorphic facial signs for medical diagnosis
- Biometrics / face recognition



[1] https://www.kaggle.com/c/facial-keypoints-detection/

# Data

Dataset provided via Dr. Yoshua Bengio, University of Montreal

1. Each example contains:
    a. Unique identification code
    b. Image stored as 96 x 96 pixel arrays (9216 features)
    c. 30 key point features
        i. 15 pairs of X and Y's
2. Training Examples - 7,049
3. Test Examples - 1,783

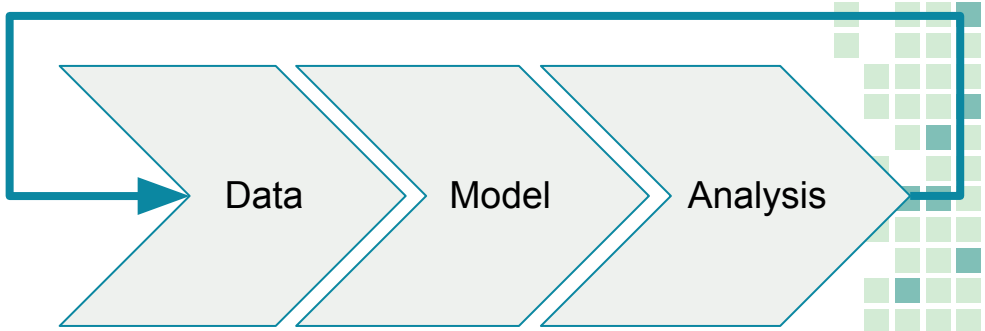
| Feature Pair (x , y) | Percent Missing |
|---|---|
| right_eyebrow_outer_end | 69% |
| left_eye_inner_corner | 68% |
| left_eye_outer_corner | 68% |
| right_eye_inner_corner | 68% |
| right_eye_outer_corner | 68% |
| left_eyebrow_inner_end | 68% |
| left_eyebrow_outer_end | 68% |
| right_eyebrow_inner_end | 68% |
| mouth_left_corner | 68% |
| mouth_right_corner | 68% |
| mouth_center_top_lip | 68% |
| mouth_center_bottom_lip | 0.4% |
| right_eye_center | 0.2% |
| left_eye_center | 0.1% |
| nose_tip | 0.0% |

# Modeling strategy

1. Baseline Model

2. Cascade model
   a. Phase 0: OLS
   b. Phase 1: DNN
   c. Phase 2: CNN
   d. Phase 3: CNN with transfer learning

Loss: RMSE




Understanding data characteristics

Augmenting data with more samples

Creating more samples for misclassified images

Comparison of architectures

Model architecture optimization

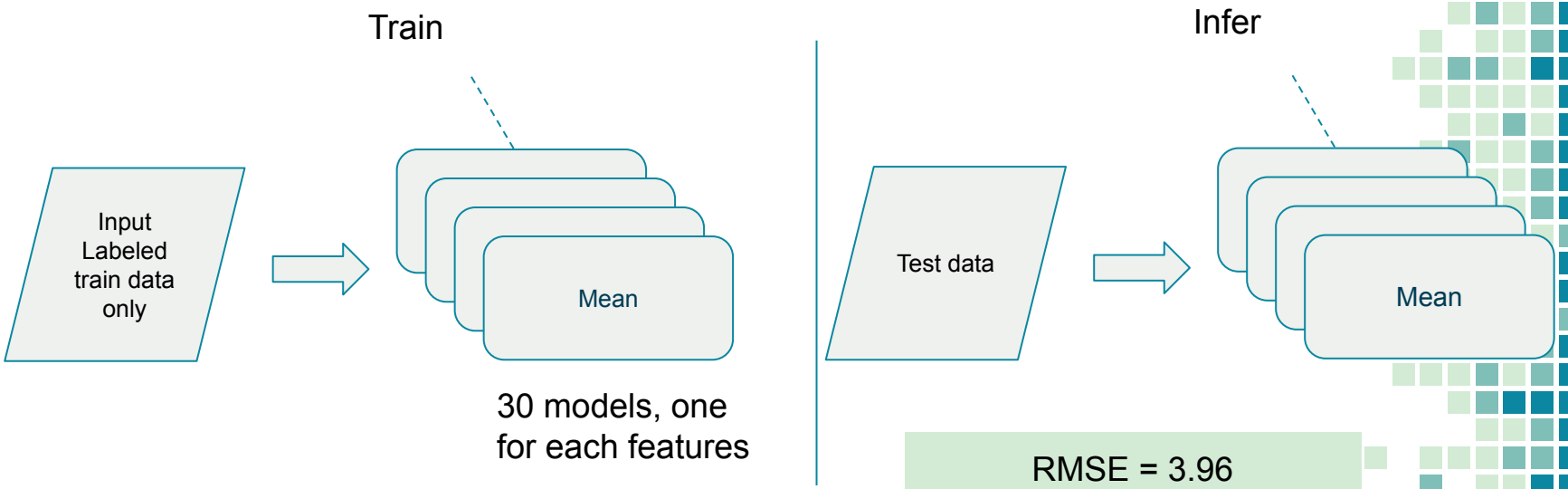
Hyperparameter optimization

Model dashboard

Visualizing misclassifications

Root-cause analysis

# Baseline - Location Means




RMSE = 3.96

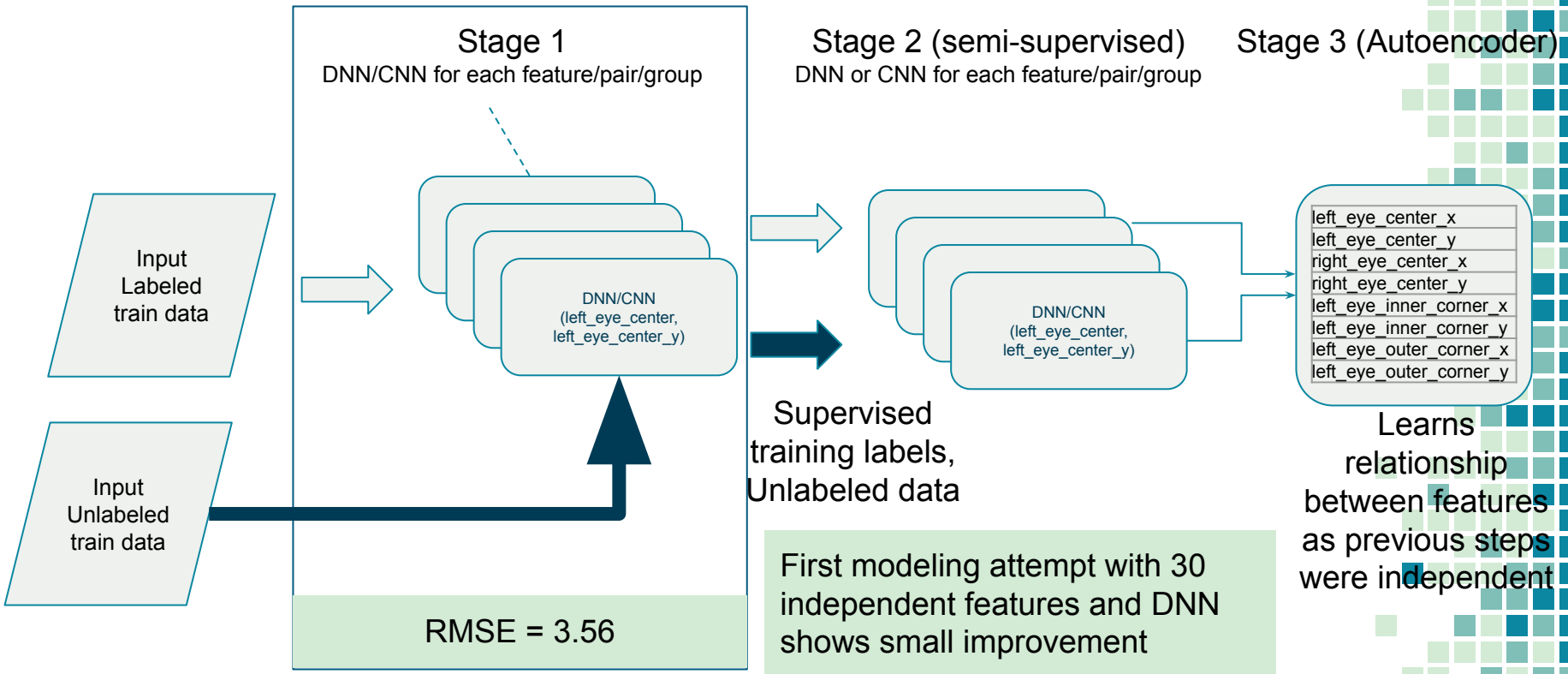
Your most recent submission

| Name | Submitted | Wait time | Execution time | Score |
|---|---|---|---|---|
| MySubmission.csv | just now | 1 seconds | 1 seconds | 3.96243 |

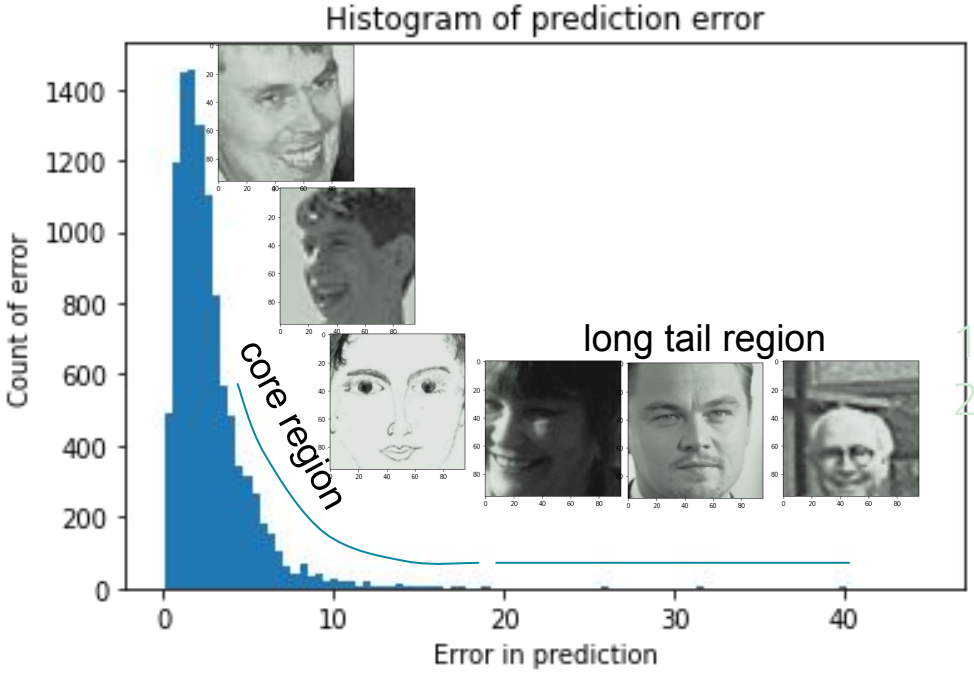
Complete

Jump to your position on the leaderboard

# Cascade model



# Analysis of predictions (Stage 1: DNN)




1. Long tail region
2. Core region

# Next steps

# Data Augmentation

- Model performance improves with increased training data
- Challenge: Limited training data w.r.t. number of features to train.
- Solution: Create new training examples via data augmentation

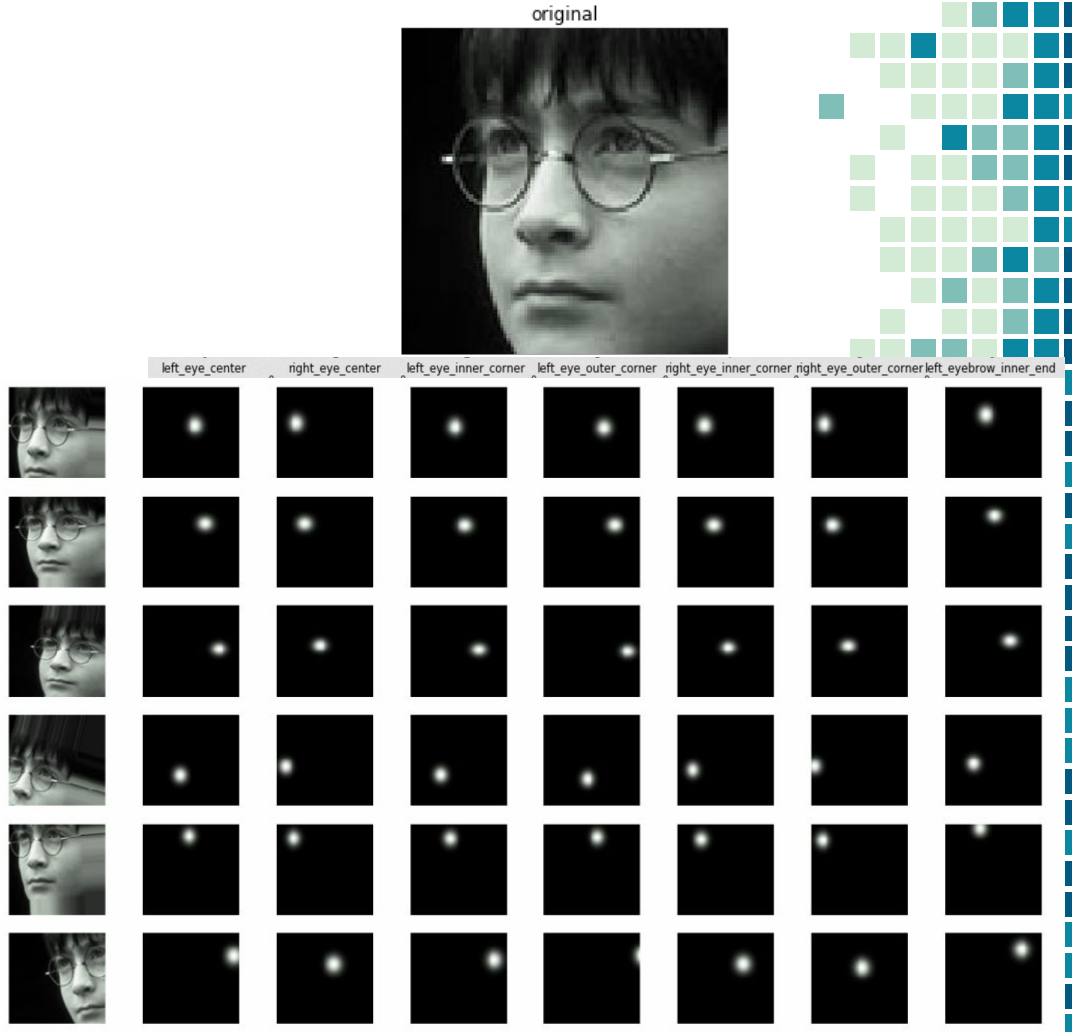
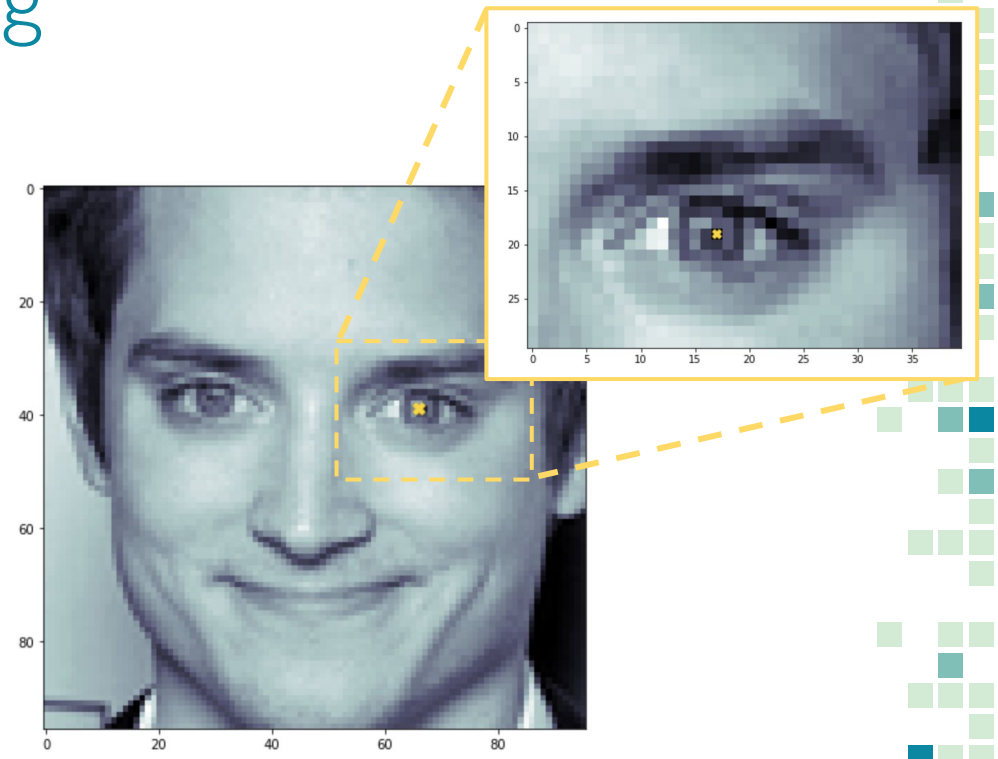



Image Source: https://fairyonice.github.io/Achieving-top-5-in-Kaggles-facial-keypoints-detection-using-FCN.html

# Image Preprocessing

- Models perform better with less noise and features
- Challenge: Lots of unnecessary features assessed by each model
- Solution: Crop Images for each feature to reduce number of features to train



# Transfer Learning

- Model Parameters are Critical
- Challenge: Training each CNN is very time consuming
- Solution: Leverage pretrained existing models via transfer learning

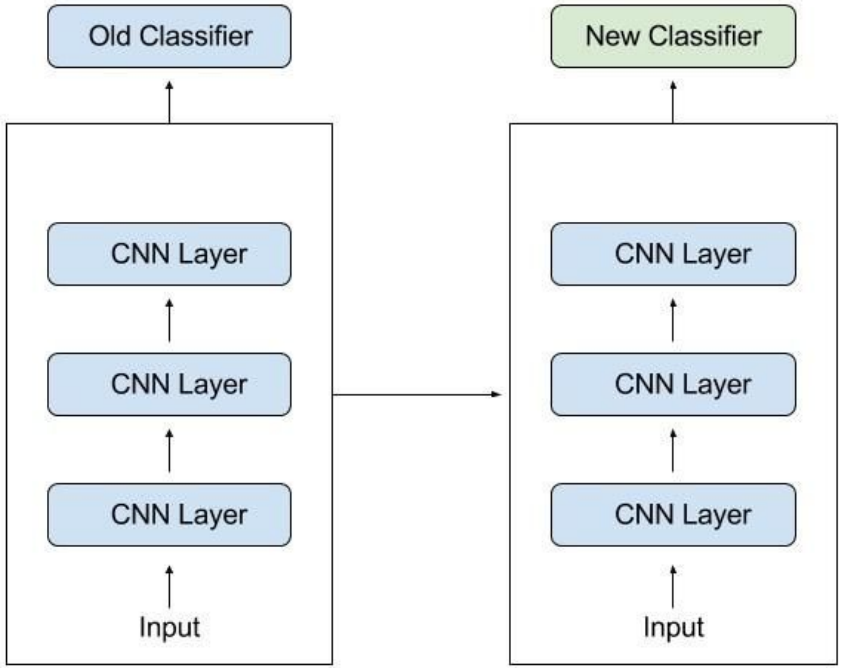



Image Source: https://builtin.com/data-science/transfer-learning

# Tasks

Data exploration

- Sweta and Leon to look at

Operating model

- Final submission in notebook
  - Only add final(or working) version to the main submission notebook
- Create functions in your own notebook or import as a py file
- Store all work in Git (including your own work-in-progress code too)

# Reference

Augmentation

- https://journalofbigdata.springeropen.com/articles/10.1186/s40537-019-0197-0
- https://fairyonice.github.io/Data-augmentation-for-facial-keypoint-detection.html

- GANN for increasing training data on misclassified samples

# Model

Model dashboard

- Learning curves

Model architecture and hyperparameter optimization

- Genetic algo optimization?

Transfer learning

Batch normalization

Dropout

# Thank you!