# Project A: Person Identification from Face Images

### CSC 481/581

### Artificial/Computational Intelligence

### Summer 2018

## Introduction

In this project our goal is to identify persons from face images. Using the AR Face Database we will extract features of eye length, eye distance, nose size, lip size, lip length, eye-brow length, and aggressive ratio. We will experiment with K-Nearest Neighbor, Artificial Neural Network and Naïve Bayes to find the accuracy, precision and recall rate of the classifiers predictions.

## Problem Description

To be added.

## Data Sets and Experiment Design

To be added.

## Defining Features and Feature Extraction

### Defining Features

We started with the following 7 features: eye length ratio, eye distance ratio, nose ratio, size ratio, lip length ratio, eye-brow length ratio, aggressive ratio.

### Feature Extraction

To be added.

## Result and Analysis

### Confusion Matrix

* K-Nearest Neighbors
  + Precision: ~
  + Recall Rate: 0.6
  + Accuracy: 0.2
  + Confusion Matrix:

[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]

[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]

[0 0 1 0 0 0 0 0 0 0 0 0 0 0 0]

[0 0 0 0 0 0 1 0 0 0 0 0 0 0 0]

[0 0 0 1 0 0 0 0 0 0 0 0 0 0 0]

[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]

[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]

[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]

[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]

[0 0 0 0 0 0 0 0 0 1 0 0 0 0 0]

[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]

[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]

[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]

[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]

[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]

* Artificial Neural Network
  + Precision:
  + Recall Rate:
  + Accuracy:
* Naïve Bayes
  + Precision:
  + Recall Rate:
  + Accuracy:

## Conclusion

To be added.

## References

*A.M. Martinez and R. Benavente. The AR Face Database. CVC Technical Report #24, June 1998.*

[*https://machinelearningmastery.com/rescaling-data-for-machine-learning-in-python-with-scikit-learn/*](https://machinelearningmastery.com/rescaling-data-for-machine-learning-in-python-with-scikit-learn/)

*http://sebastianraschka.com/Articles/2014\_about\_feature\_scaling.html*

*http://scikit-learn.org/stable/datasets/index.html*

[*https://towardsdatascience.com/train-test-split-and-cross-validation-in-python-80b61beca4b6*](https://towardsdatascience.com/train-test-split-and-cross-validation-in-python-80b61beca4b6)

*http://scikit-learn.org/stable/modules/preprocessing.html#preprocessing*

[*https://www.ritchieng.com/machine-learning-k-nearest-neighbors-knn/*](https://www.ritchieng.com/machine-learning-k-nearest-neighbors-knn/)

[*https://kevinzakka.github.io/2016/07/13/k-nearest-neighbor/*](https://kevinzakka.github.io/2016/07/13/k-nearest-neighbor/)

[*http://scikit-learn.org/stable/auto\_examples/neighbors/plot\_classification.html#sphx-glr-auto-examples-neighbors-plot-classification-py*](http://scikit-learn.org/stable/auto_examples/neighbors/plot_classification.html#sphx-glr-auto-examples-neighbors-plot-classification-py)

[*http://scikit-learn.org/stable/auto\_examples/preprocessing/plot\_scaling\_importance.html#sphx-glr-auto-examples-preprocessing-plot-scaling-importance-py*](http://scikit-learn.org/stable/auto_examples/preprocessing/plot_scaling_importance.html#sphx-glr-auto-examples-preprocessing-plot-scaling-importance-py)

[*http://scikit-learn.org/stable/modules/generated/sklearn.metrics.confusion\_matrix.html*](http://scikit-learn.org/stable/modules/generated/sklearn.metrics.confusion_matrix.html)

*https://www.ritchieng.com/machine-learning-k-nearest-neighbors-knn/*

## Appendices

To be added.