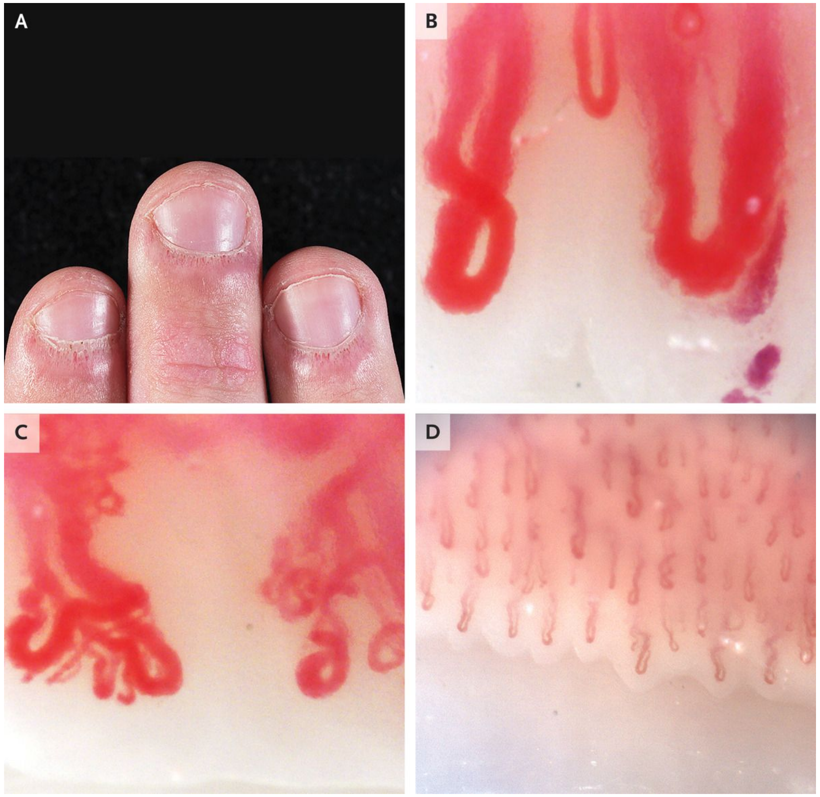
University of California, Irvine

CHOC1

Tarek El-Hajjaoui, Micah Fadrigo, Sheldon Gu, Cecilia Nguyen

- Juvenile dermatomyositis (JDM)
- Rare
- No cure → **improve disease management**
- Expensive labs

Project Description



Nailfold Capillaroscopy (NFC)

GOAL

Simple, quick & inexpensive pre-screening on Juvenile Dermatomyositis (JDM)

1. **Data**
2. Machine Learning Models
    a. Logistic Regression + Lasso Model
    b. Random Forest
    c. Support Vector Machine
3. Next Steps

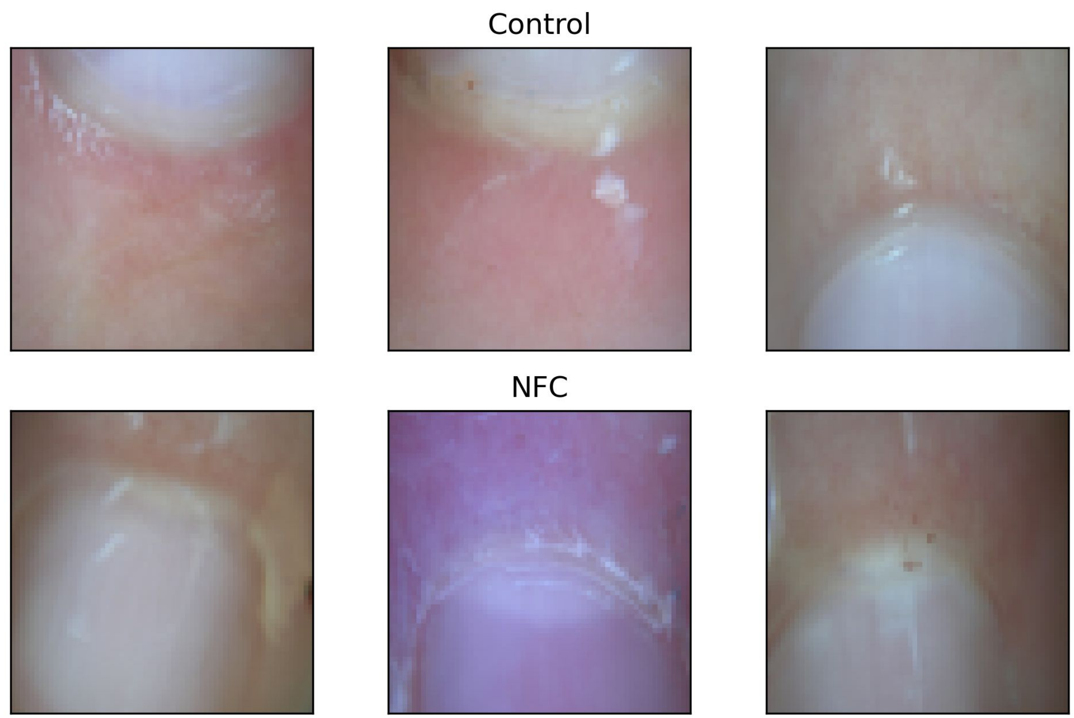
Data Description

Image Level:

- JDM: **1120** images
- Control: **321** images

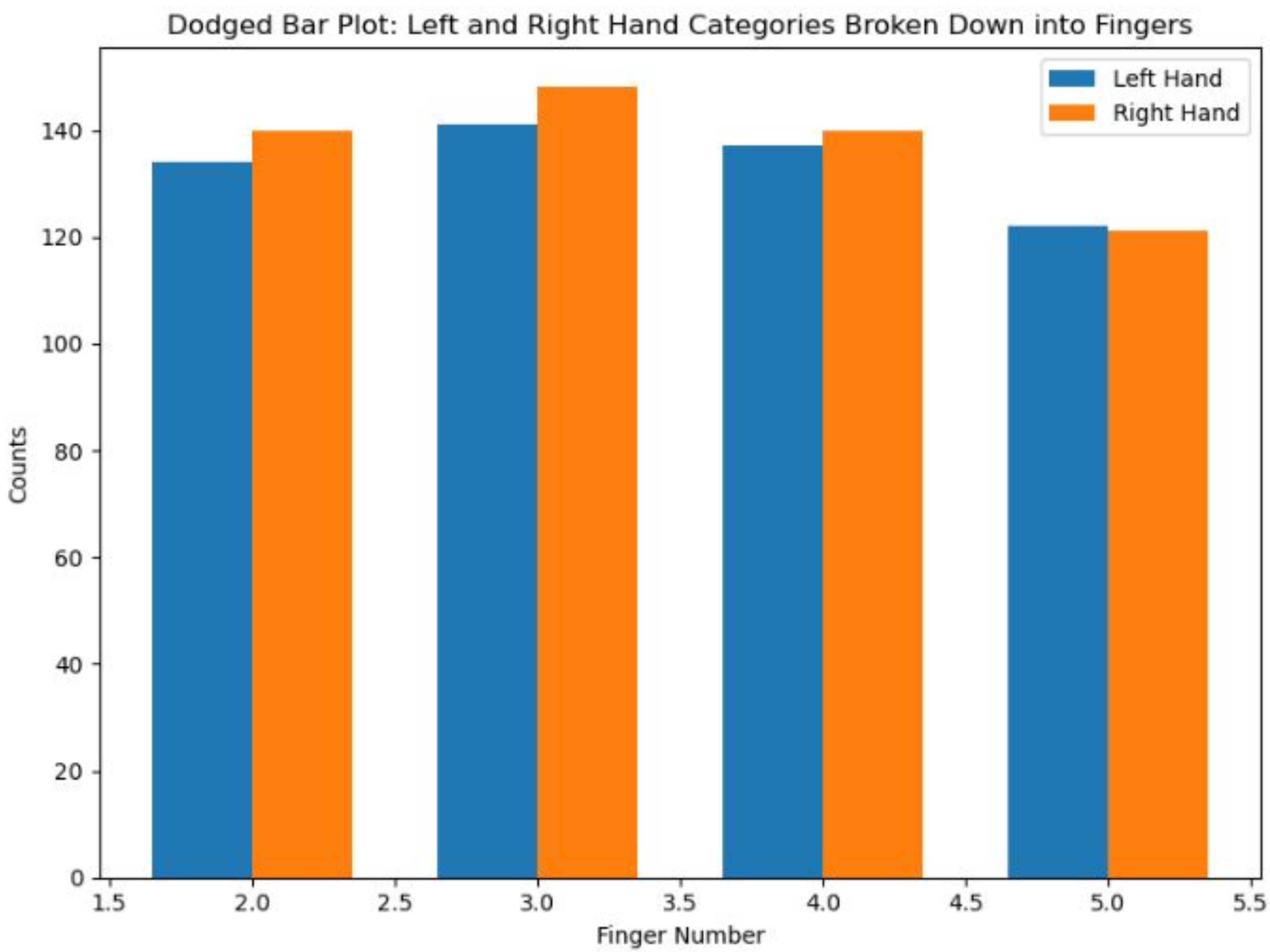
Patient Level:

- JDM: **111** patients
- Control: **31** patients

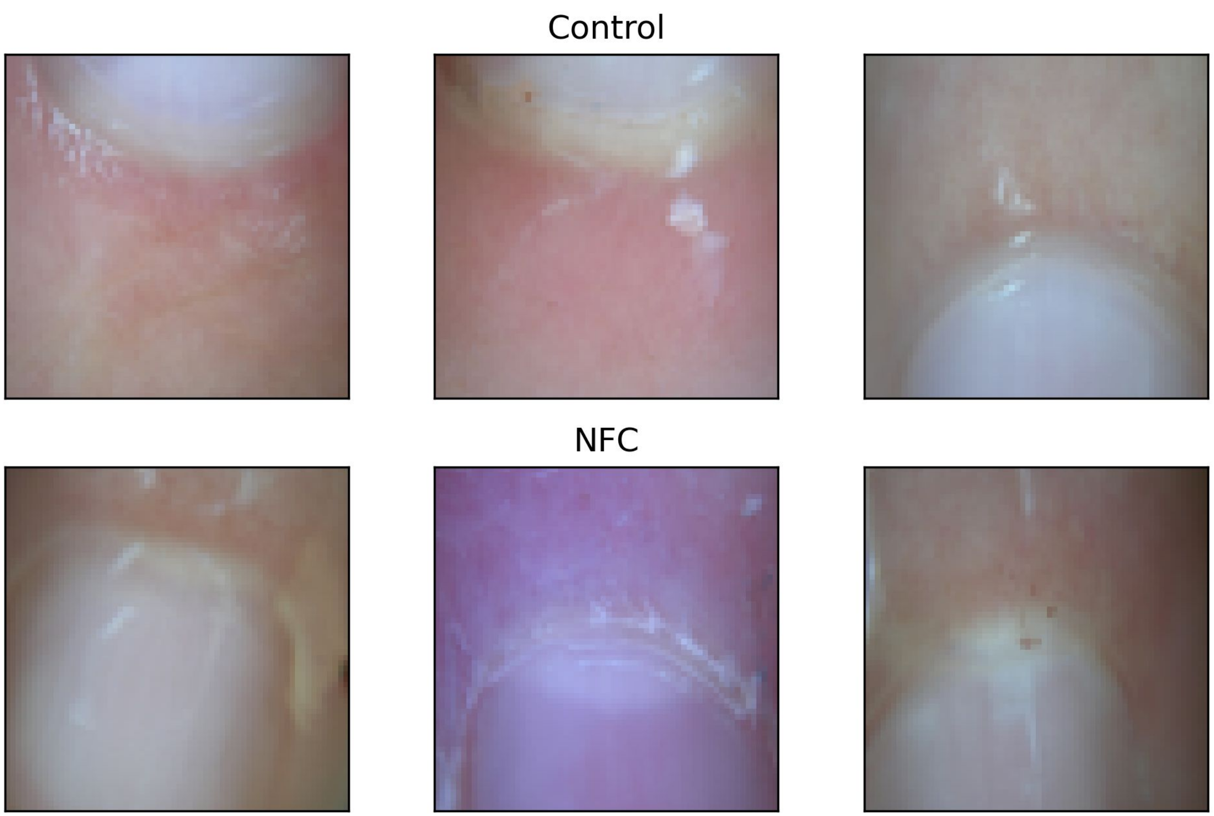


Response variable: JDM & Control

EDA


Dodged Bar Plot: Left and Right Hand Categories Broken Down into Fingers
Left Hand
Right Hand
Counts
0
20
40
60
80
100
120
140
1.5
2.0
2.5
3.0
3.5
4.0
4.5
5.0
5.5
Finger Number

Data Issue Beyond Control



- **Unclear** (reflected lights, too dark)
- **Skin tone** varies

Data Issue Beyond Control

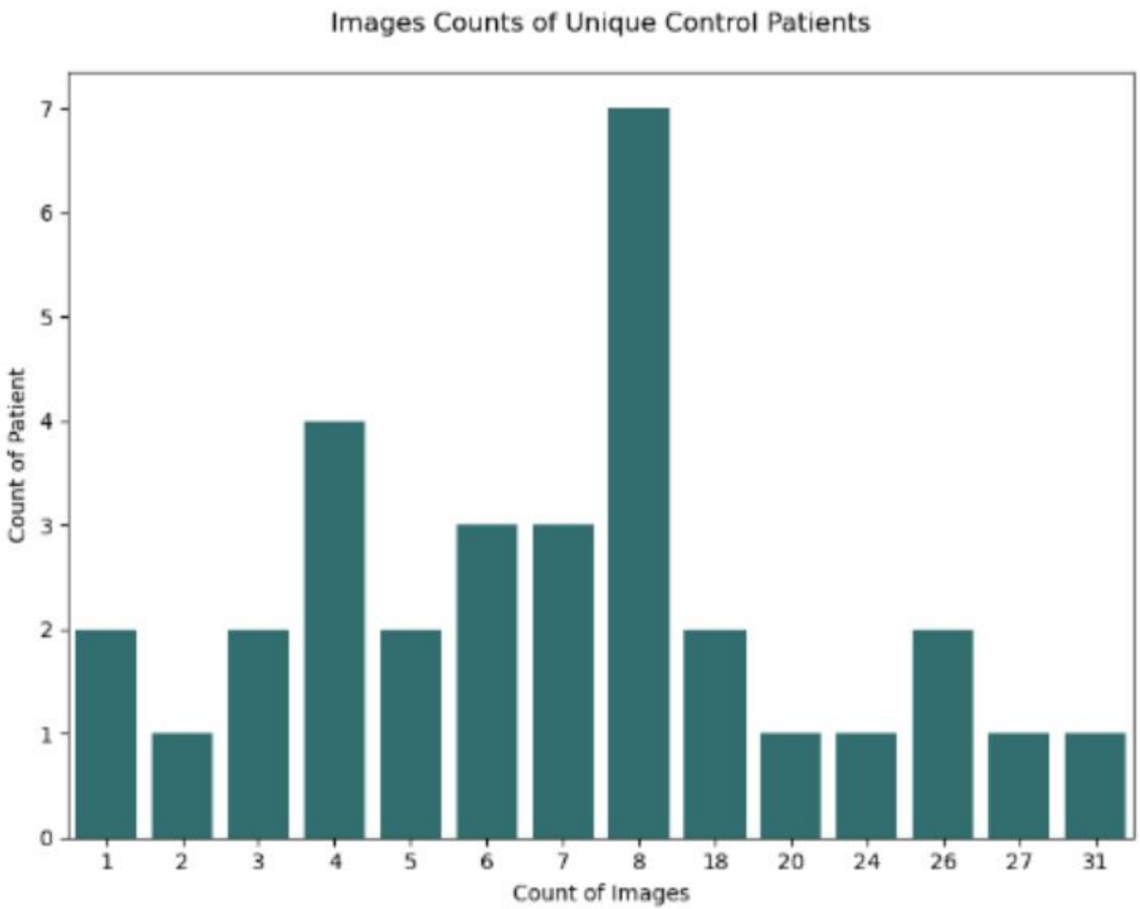
Obstructed Images

https://www.stylecraze.com/articles/8-simple-nail-art-desi
gns/

https://laurenbbeauty.com/blogs/blog/how-to-remove-nail-
polish-from-skin-around-nails

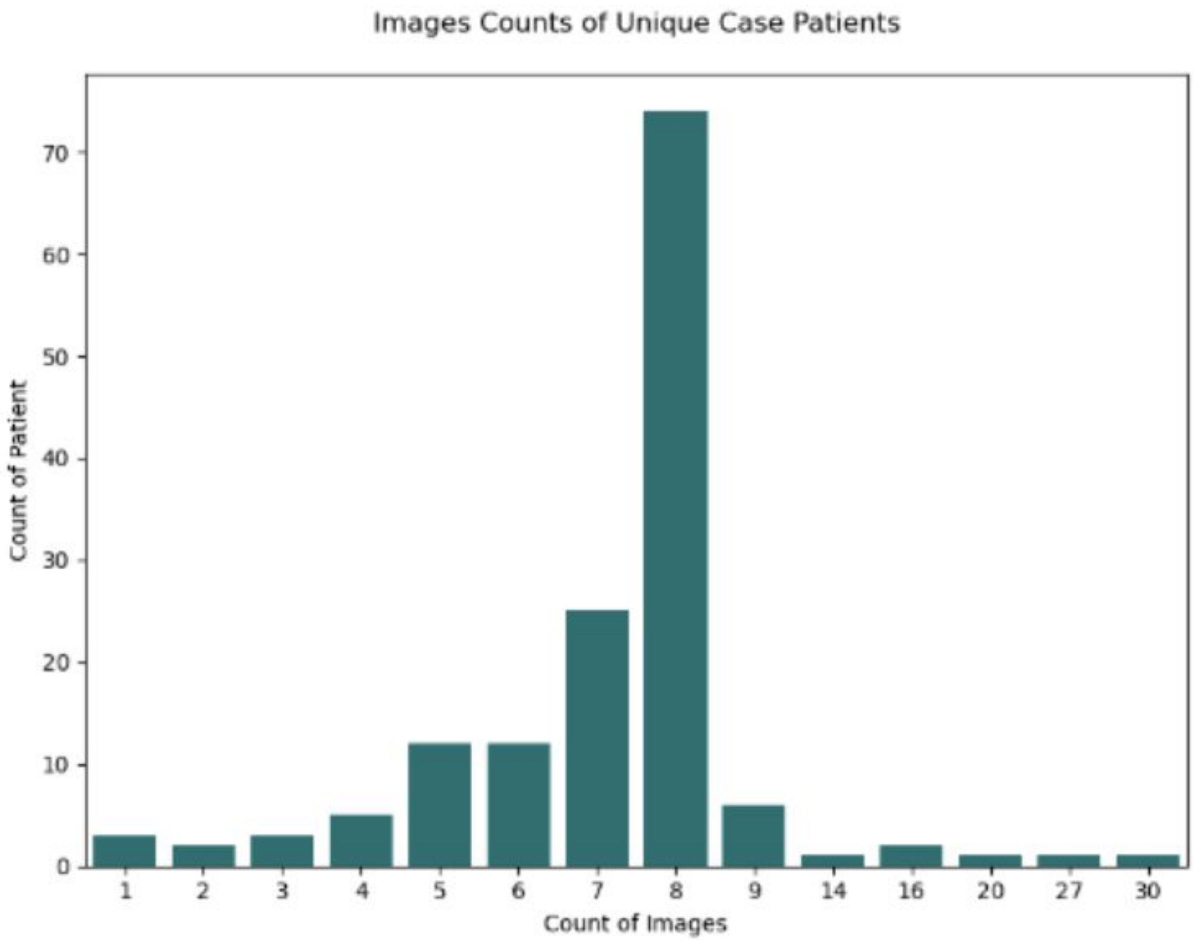
Data Issue Beyond Control

Imbalance: Not all patients represented equally between case/control and within.

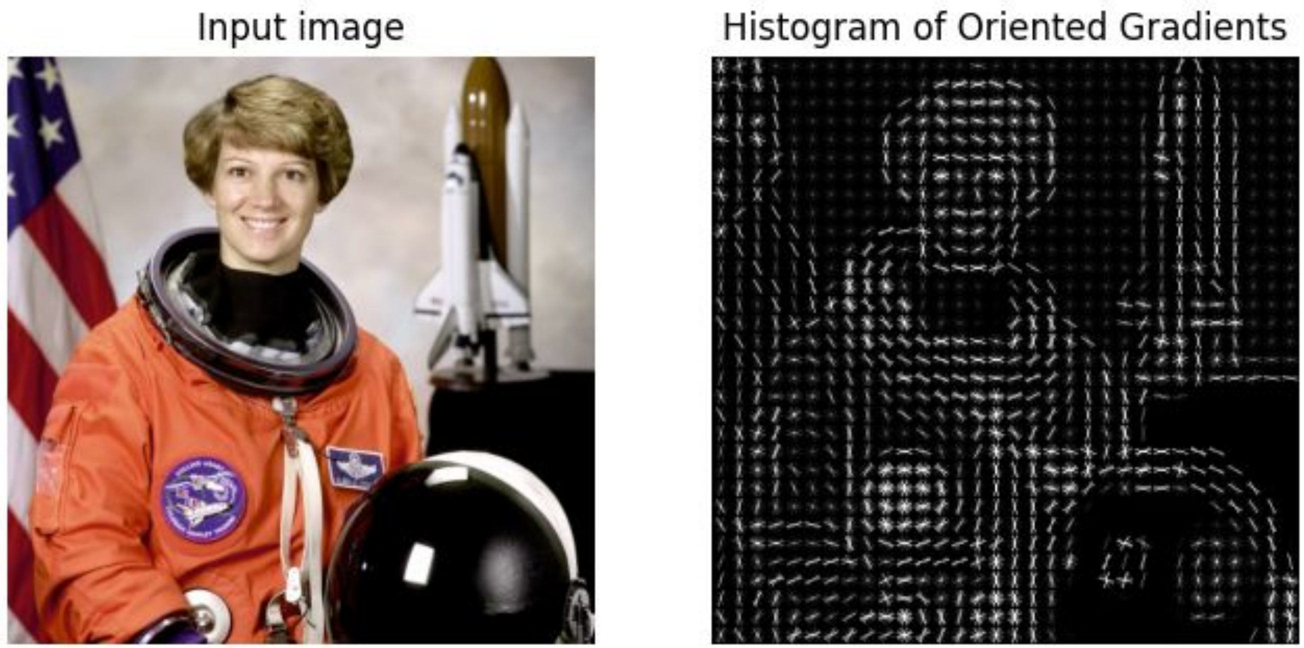


Data Issue Beyond Control

Imbalance: Not all patients represented equally between case/control and within.

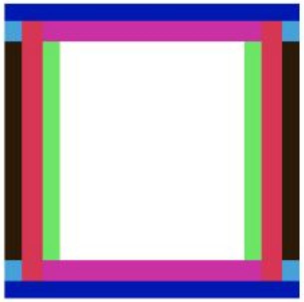


Data Issue - Solution

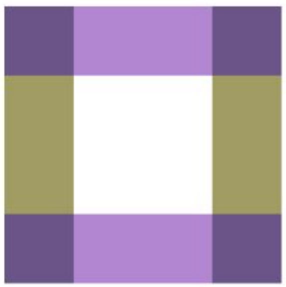


- **Large image size** → Interpolation
- **Absence of feature** → Histogram of Oriented Gradients (HOG)

Data Preprocess



initial image
[16px x 16px]



PIL.Image.resize
[4px x 4px]

tf.image.resize_bicubic
[4px x 4px]

CHOC: Manually examined and corrected orientation of NFCs.

Our Steps:

1. **Downscale** to size: 128, 64, 32
2. **Scale** input pixels between **(-1, 1)**
3. **Vectorize** images
4. **HOG** transformation
5. 10-Fold Stratified **Cross-Validation**

Histogram of Oriented Gradients (HOG)

What is HOG

- Computer vision feature descriptor technique.
- Distribution of edge orientations.

Why is this useful

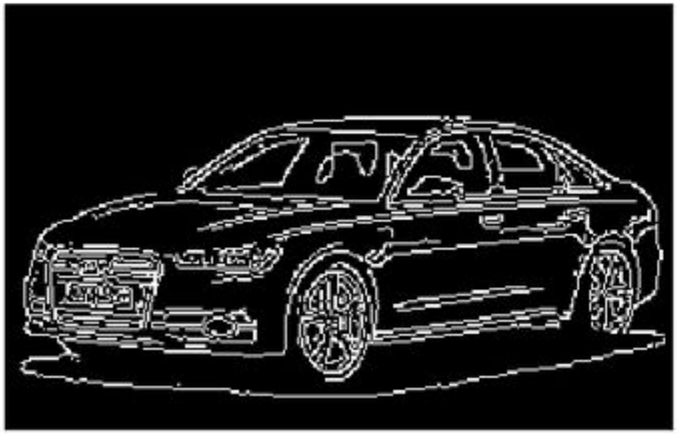
- Learn structural and spatial patterns of images.
- Reduces noise of images (for classification or object detection tasks).
- Generally preferred over vectorized images.

HOG Example

Which features of these images can be used to differentiate these objects?

HOG Example

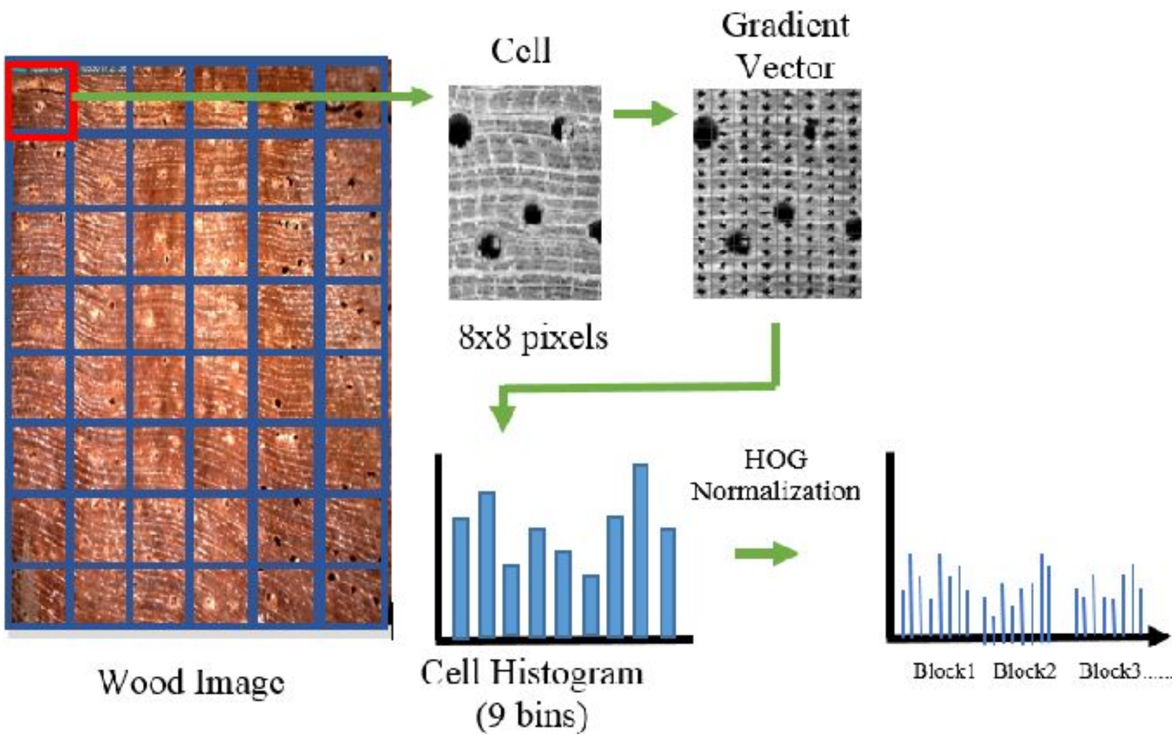
- The objects can distinguished using only their shapes & edges.
- No need for color, background, etc.



HOG Explained

How HOG works

- Gradient magnitude & orientation is computed for each pixel in an image.
- Similar to a Convolutional layer, the image is divided into smaller cells.



HOG Example

Example from scikit-image (library used)

Input image

Histogram of Oriented Gradients

1. Data
2. **Machine Learning Models**
   a. Logistic Regression + Lasso Model
   b. Random Forest
   c. Support Vector Machine
3. Next Steps

Convolution Neural Networks

- **Widely used** for computer vision tasks

- Standard Architectures:
  - Batch normalization is **sensitive to large variation in the data**
  - **Uninterpretable**

- CHOC developed NFC-Net = **lightweight** CNN = 3 layers
  - Working on explainability

	Accuracy	Precision	Recall	F1 Score	ROC AUC	Specificity
NFC-Net	0.91	0.95	0.85	0.897	0.93	0.9

Overall Accuracy

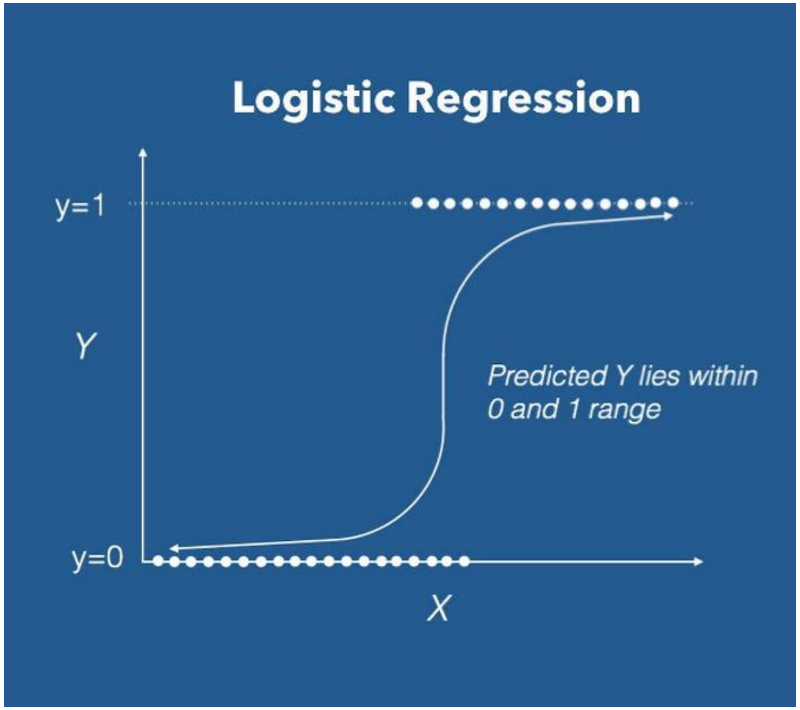
Class 1 Accuracy

Class 0 Accuracy

Why Pursue Simpler Models?

- ★ Baseline Measurement & Reference
  - ○ Are simple models able to achieve similar scores to NFC-Net?
- ★ Quicker Deployment to Mobile Devices
  - ○ **Automate** clinical analyses of NFC
  - ○ **Accelerate** JDM data collection & research
- ★ Robustness
  - ○ Deals with high-level of noise

Logistic Regression + Lasso Regularization



- Simple baseline for model for binary classification task
- Predict probability of JDM given NFC image

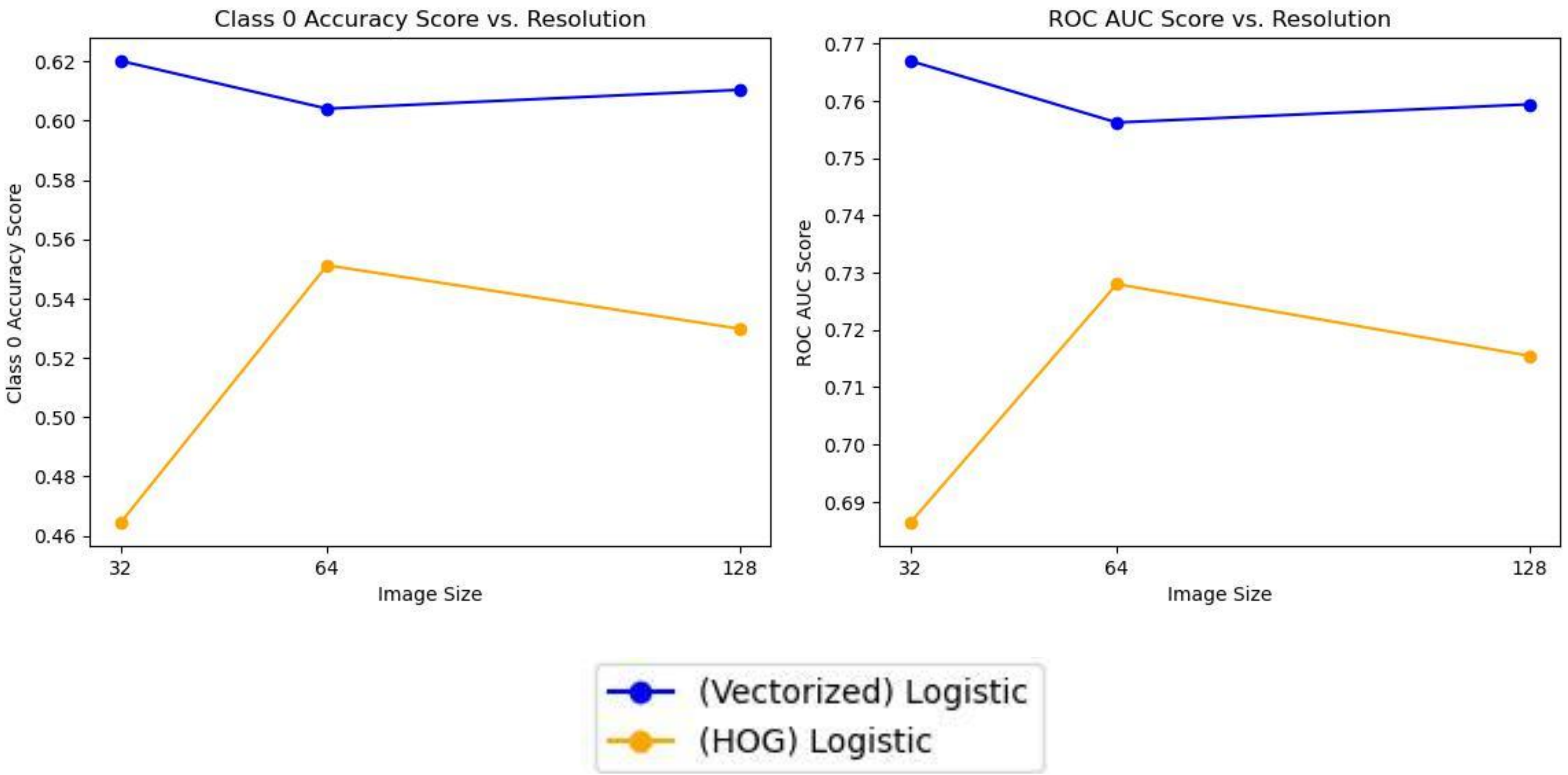
- Automatic feature selection in high-dimensional spaces
- Simplifies model → improves interpretability

Logistic Regression + Lasso Results

Complexity

	Accuracy	Precision	Recall	F1 Score	ROC AUC	Class 0 Accuracy
(Vectorized) Logistic [32x32]	0.848	0.893	0.914	0.903	0.767	0.62
(HOG) Logistic [32x32]	0.809	0.855	0.908	0.881	0.686	0.464
(Vectorized) Logistic [64x64]	0.84	0.888	0.908	0.898	0.756	0.604
(HOG) Logistic [64x64]	0.826	0.876	0.905	0.889	0.728	0.551
(Vectorized) Logistic [128x128]	0.842	0.89	0.908	0.899	0.759	0.61
(HOG) Logistic [128x128]	0.818	0.869	0.901	0.885	0.715	0.53

Logistic Regression + Lasso Results



Logistic Regression: Challenges

- Real-world settings: Assumptions may be violated
- Image data: **Non-linear relationships** between outcome & predictors
- Images within patient may yield similar risk of JDM
  ➔ Remove **highly-correlated images**
- **Highly-correlated features** undermines:
  - Model interpretability
  - Reliability of coefficient estimate
  - Statistical significance of features

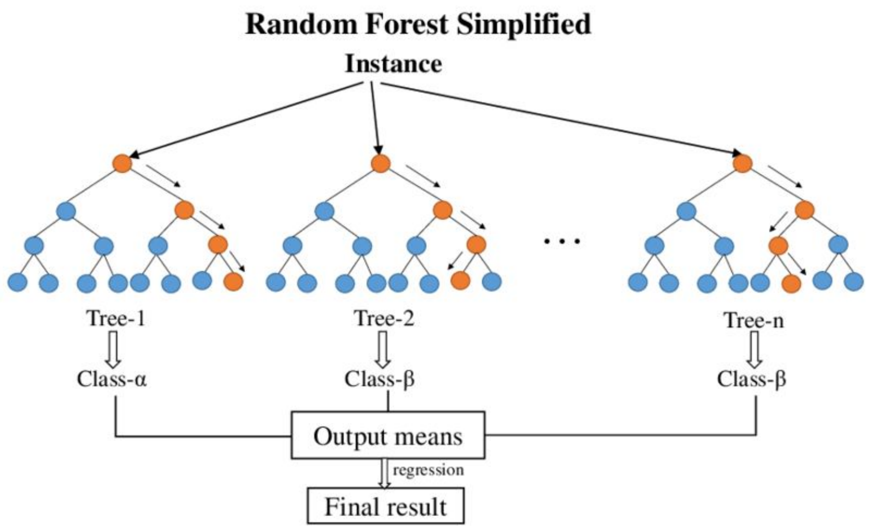
  ➔ Lasso regularization
  ➔ Variance Inflation Factor

★ **May not be suitable approach for this problem**

UCI

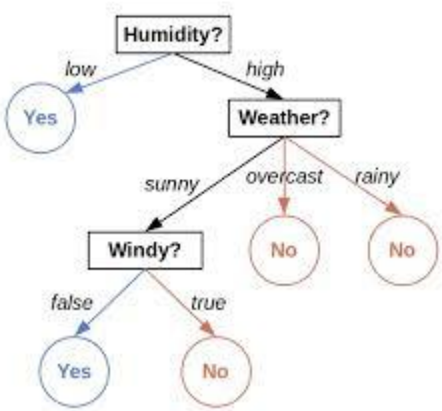
Random Forest

- An ensemble of decision trees, in which randomly selected subsets of data are trained in each decision tree



<u>Why Random Forest?</u>

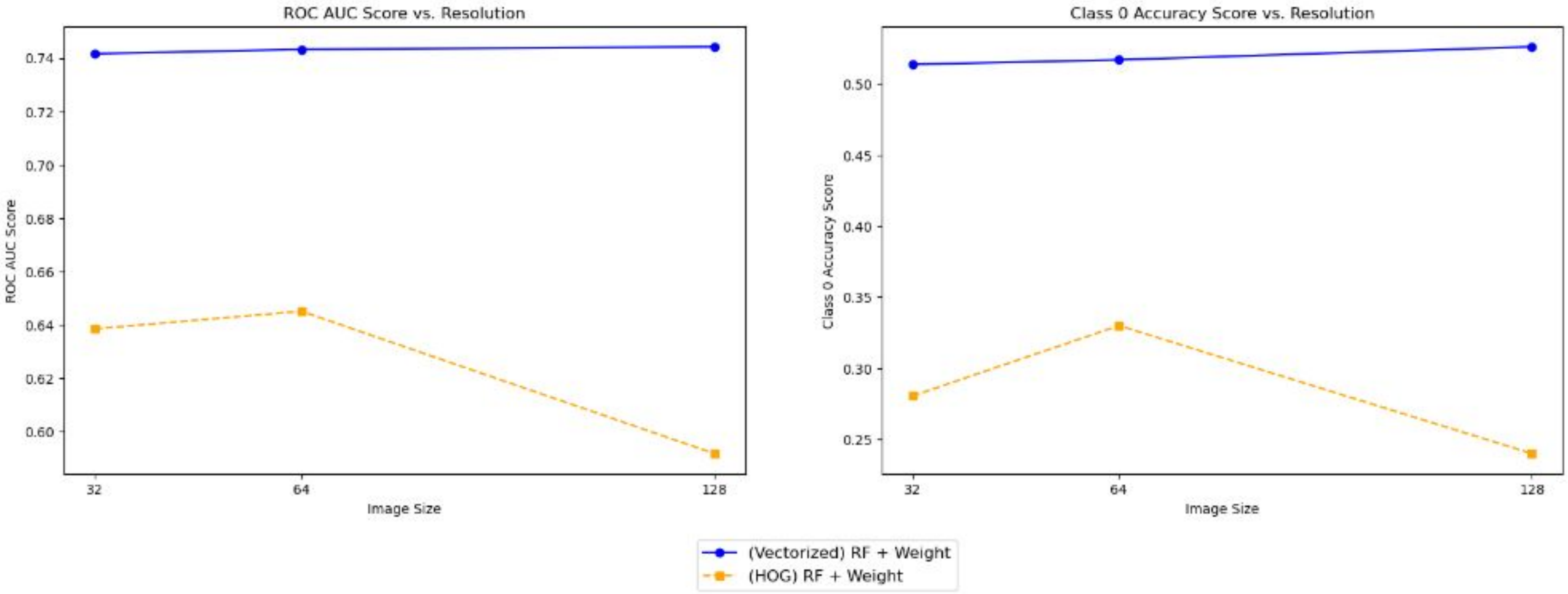
- Classification Model
- No assumptions underlying distribution
- Robust to Overfitting
- Might capture complex relationships in data, works with non-linear data
- Robust to outliers/noises in training data



Random Forest Results

Weight (1.5:1)	Accuracy	Precision	Recall	F1 Score	ROC AUC	Class 0 Accuracy
(Vectorized) RF [32x32]	0.869	0.877	0.968	0.920	0.748	0.529
(HOG) RF [32x32]	0.824	0.818	0.995	0.89	0.614	0.234
(Vectorized) RF [64x64]	0.869	0.874	0.971	0.920	0.743	0.514
(HOG) RF [64x64]	.839	0.832	0.993	0.905	0.649	0.305
(Vectorized) RF [128x128]	0.863	0.872	0.966	0.916	0.735	0.504
(HOG) RF [128x128]	0.819	0.814	0.994	0.895	0.603	0.212

Random Forest Results



Random Forest Challenges

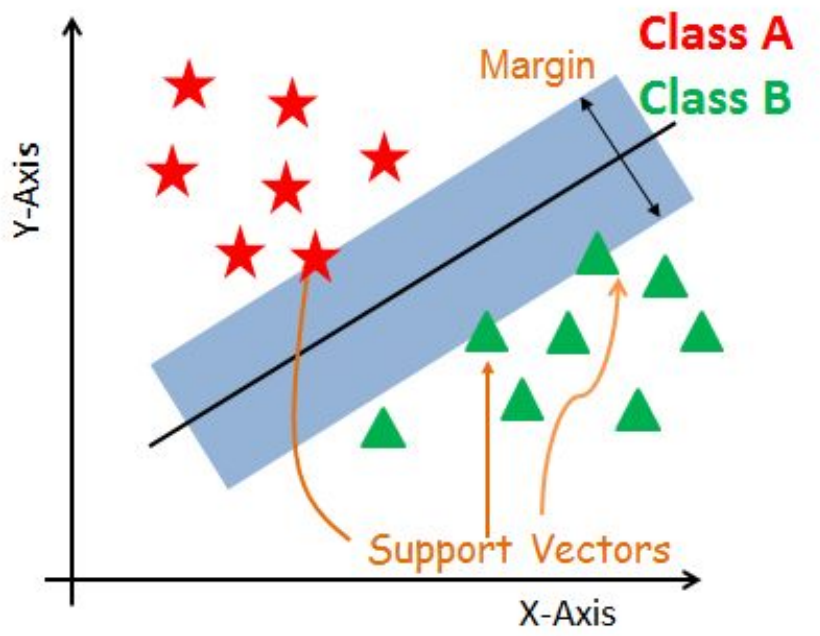
Why are we not pursuing to optimize this model?

- Low overall performance
- Not as explainable for spatial data
  - Determine which pixels important
- Inefficient Computational efficiency, high-dimensional data

Support Vector Machine (SVM)

SVM Main Concepts

- Decision Boundary, Margins, Support Vectors & Kernel.



Why SVM?

- Suits binary classification.
- SVM with HOG is proven effective for computer vision.
- Computationally efficient.
- Robust to overfitting.

SVM with Linear Kernel Function Results

Complexity

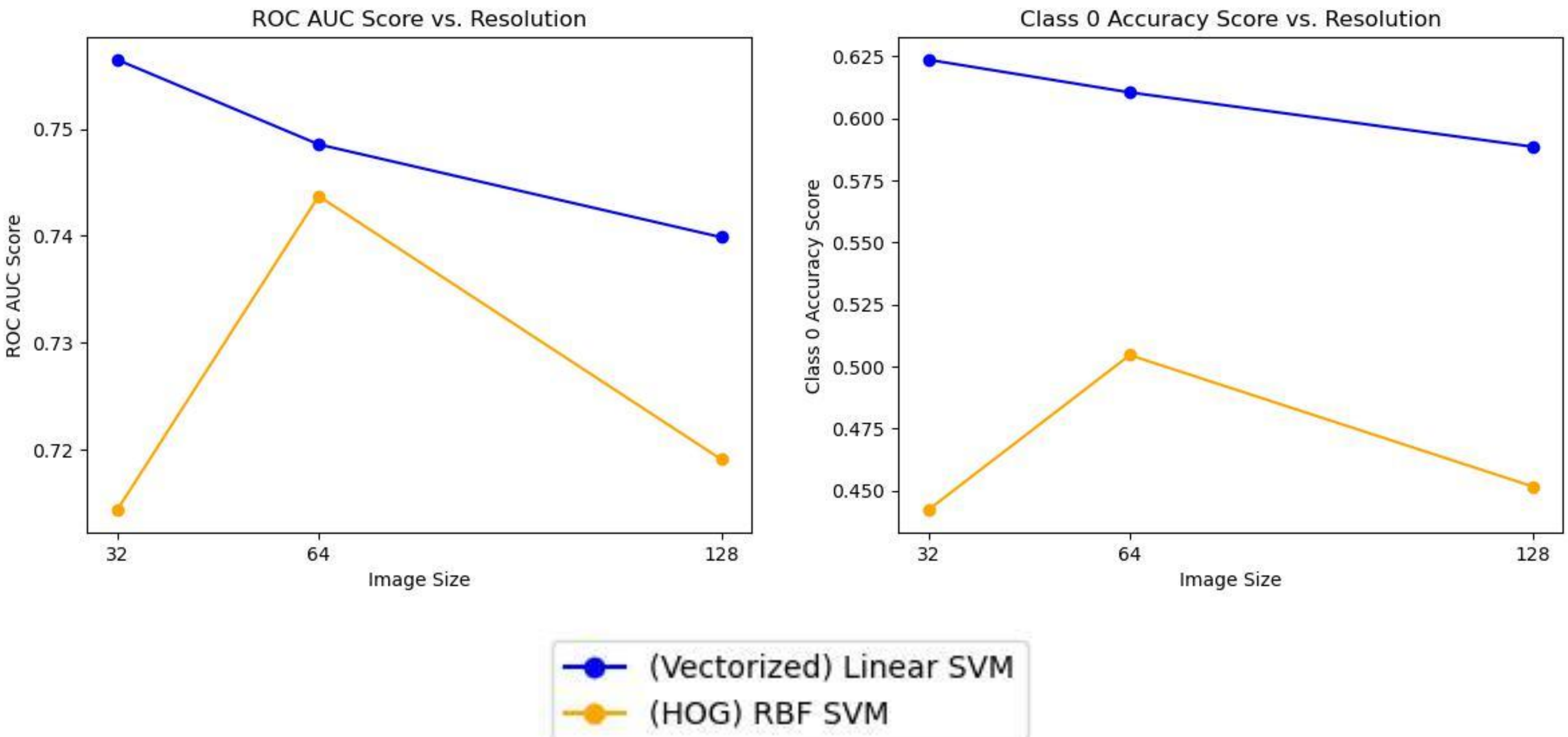
	Accuracy	Precision	Recall	F1 Score	ROC AUC	Class 0 Accuracy
(Vectorized) SVM Linear [32x32]	0.830	0.892	0.889	0.890	0.756	0.624
(HOG) SVM Linear [32x32]	0.807	0.838	0.933	0.882	0.653	0.374
(Vectorized) SVM Linear [64x64]	0.825	0.888	0.887	0.887	0.749	0.610
(HOG) SVM Linear [64x64]	0.809	0.876	0.880	0.878	0.722	0.564
(Vectorized) SVM Linear [128x128]	0.823	0.883	0.891	0.887	0.74	0.589
(HOG) SVM Linear [128x128]	0.828	0.885	0.895	0.890	0.745	0.595

SVM with RBF Kernel Function Results

Complexity

	Accuracy	Precision	Recall	F1 Score	ROC AUC	Class 0 Accuracy
(Vectorized) SVM RBF [32x32]	0.853	0.858	0.971	0.911	0.707	0.442
(HOG) SVM RBF [32x32]	0.865	0.860	0.987	0.919	0.714	0.442
(Vectorized) SVM RBF [64x64]	0.852	0.858	0.971	0.911	0.705	0.439
(HOG) SVM RBF [64x64]	0.876	0.873	0.983	0.925	0.744	0.505
(Vectorized) SVM RBF [128x128]	0.852	0.858	0.971	0.911	0.705	0.439
(HOG) SVM RBF [128x128]	0.867	0.862	0.987	0.920	0.719	0.452

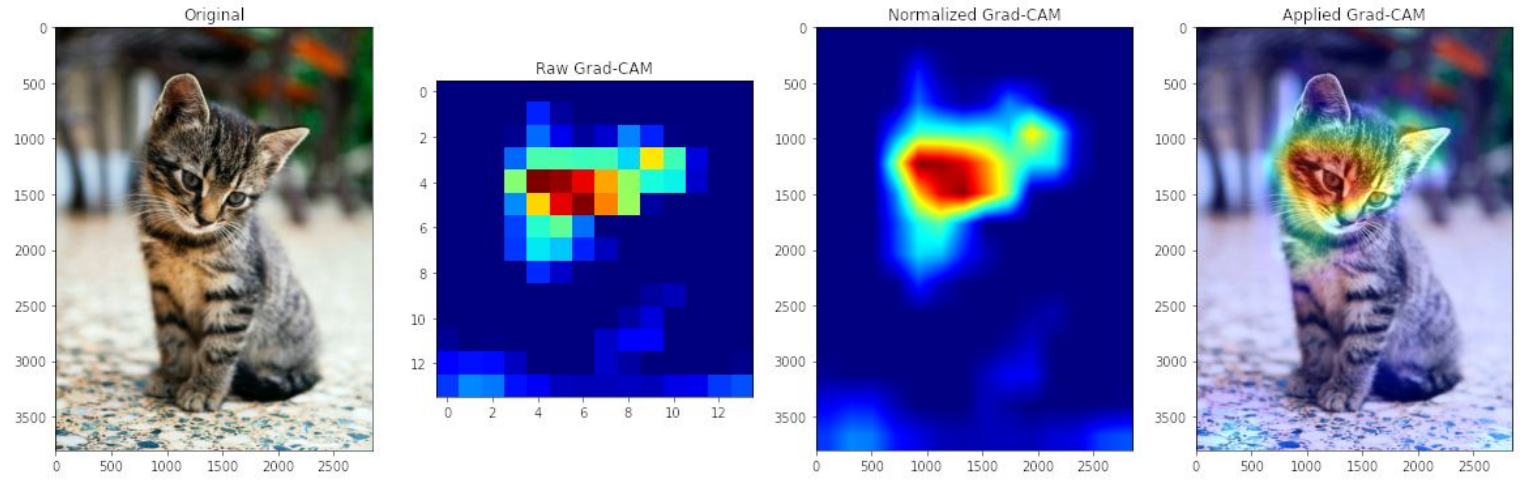
SVM Results (Cont)



SVM Improvements

- HOG tuning: orientations, pixels_per_cell, & cells_per_block
- More dimension sizes: 16 & 256
- Model Explainability

Goal: create an equivalent of CNN explainability but for Linear SVM models:



1. Data
2. Machine Learning Models
    a. Logistic Regression + Lasso Model
    b. Random Forest
    c. Support Vector Machine
3. **Next Steps**

Next Steps

- **Building and improve CNN**
  - hyperparameter
  - different activation
  - different constructions
- **Explainable Neural Network**
- **Image Augmentation**

Timeline

6	7	8	9	More
Data Process	**Models**	**Improve CNN**	**Final**	**Other Techniques**
EDA	HOG lecture	Different construction	Conclusion	Image Augmentation
Downscale	Logistic Regression	Feature Extractions	Report + Presentation	Hyperparameter
Preliminary	SVM	Explainable NN		
	Random Forest			

Q & A