ImgKnock: Novel Knockoff Inference for Image Data via Latent Representation Learning

Zhe Fei, Jericho Lawson

University of California, Riverside
Department of Statistics

Dec 14th, 2024

Overview

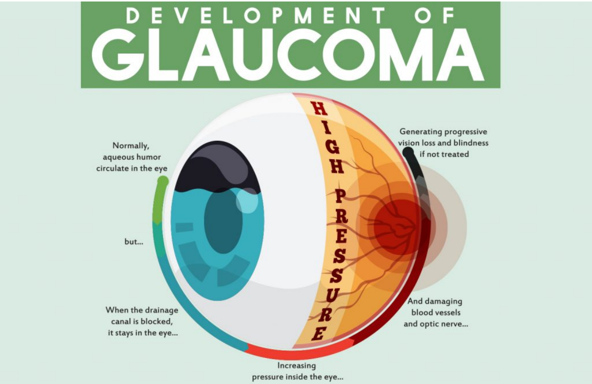
1. Motivations
   - Glaucoma Detection
   - Inference with Image Data
2. Introduction
   - Knockoffs
3. Proposed Methods
   - Notations
   - ImgKnock Procedure
4. Numerical Experiments
   - MNIST Data
   - Simulations
   - CIFAR-10 Data
   - Glacuoma Data
5. Summary
   - Next Steps

Overview

1. Motivations
   - Glaucoma Detection
   - Inference with Image Data
2. Introduction
3. Proposed Methods
4. Numerical Experiments
5. Summary

What is Glaucoma?

- **Glaucoma** is a group of eye diseases that damage the optic nerve, crucial for good vision.
- Glaucoma is one of the leading causes of blindness for people over the age of 60, second leading cause of blindness worldwide.



In the U.S.

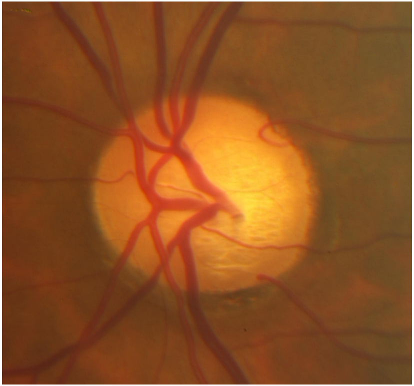
- Over 3 million Americans have glaucoma.
- It is estimated that half of them are unaware they have the disease.
- Approximately 120,000 Americans are blind from glaucoma, accounting for 9-12% of all cases of blindness.

In U.K.,

- Over 700,000 people in the UK have glaucoma.
- The number of people living with glaucoma in the UK is expected to increase by approximately 18% over the next decade.
- Glaucoma care accounts for an estimated 20% of hospital eye service outpatient workload in the UK, with over 1 million glaucoma-related outpatient visits each year.

Importance of Fundus Imaging

- **Fundus imaging** is a crucial diagnostic tool in ophthalmology for visualizing the interior surface of the eye.
- It helps in detecting and monitoring diseases such as glaucoma, diabetic retinopathy, and macular degeneration.
- The fundus includes the retina, optic disc, macula, and posterior pole.



Fundus Images in Glaucoma Diagnosis

- **Optic Disc Cupping**: Enlargement of the optic cup relative to the optic disc is a key indicator of glaucoma.
- **Neuroretinal Rim**: Thinning of the neuroretinal rim, especially at the inferior and superior poles, suggests glaucomatous damage.
- **Blood Vessel Changes**: Displacement or bending of blood vessels around the optic disc can indicate increased intraocular pressure.
- **Parapapillary Atrophy**: Loss of retinal tissue around the optic disc, known as parapapillary atrophy, is often seen in glaucoma.

Deep Learning in Glaucoma Detection

- **Automated Diagnosis**: Deep learning algorithms can automatically analyze fundus images to detect signs of glaucoma.
- **High Accuracy**: Convolutional Neural Networks (CNNs) have shown high accuracy in distinguishing between normal and glaucomatous eyes.
- **Feature Extraction**: These models can learn complex features from fundus images, such as optic disc cupping, neuroretinal rim thinning, and blood vessel patterns.
- **Early Detection**: Early detection of glaucoma through deep learning can lead to timely intervention and better management of the disease.

Take the CIFAR-10 dataset [5] for example,

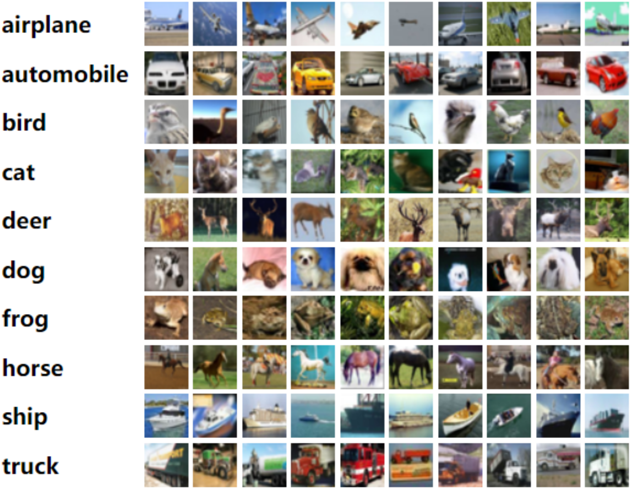


Image Classification

- Well-established problem
- **Goal**: Identify the type of image based on given image data
  - Originates from pixel data containing opacities of colors
- Examples: letters from English alphabet, main object in images, occurrence of disease

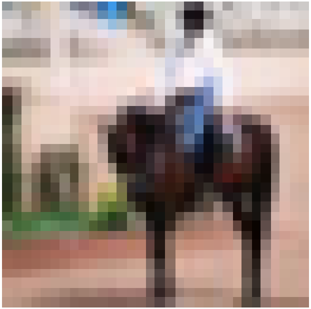



Figure: Two Images from CIFAR-10 Dataset [5]

Image Features

- Pixels could be treated as features
  - **Issues**: object of interest not always in consistent spot, data-driven
- **Solution**: Use of latent features through latent representation learning (Self-Supervised Learning)
  - **Advantages**: more data-driven, can pick up high level traits, such as optic disc cupping, neuroretinal rim thinning, and blood vessel patterns.

Overview

1 Motivations

2 Introduction
- Knockoffs

3 Proposed Methods

4 Numerical Experiments

5 Summary

Feature Importance and Selection

- Use of p-values or confidence intervals to determine significance of a feature/variable
  - Usually with assumptions on test statistic
  - Prone to p-hacking
  - Control of false discovery rate is not a guarantee
- Benjamini-Hochberg correction, Family-wise error rate
- Need a procedure that is data-driven, guarantees control of false discovery rate when looking at all variables at once, and is robust in high-dimensional settings
  - **Solution:** Knockoffs Inference

What are Knockoffs?

- Framework for controlling false discovery rate when performing variable selection, developed by Barber and Candes in 2015 [1]
  - Candes et al. (2018): model-X knockoffs [3]
  - Barber and Candes (2019): model-X knockoffs for high-dimensional linear models [2]
  - Romano, Sesia, & Candes (2019): model-X knockoffs with deep implementation [6]
- Use of negative controls
  - Synthetic variables (conditionally independent on Y given X and sharing same correlation structure)
  - Used in conjunction with original variables in variable selection procedure
  - True significance occurs when difference between original and knockoff variables exists

Knockoff Generation

For $\tilde{X}$ to be valid knockoffs of X,

i) $(X, \tilde{X}) \stackrel{d}{=} (X, \tilde{X})_{swap(S)} \quad \forall S \subset \{1, \ldots, p\}$.

ii) $X \perp \tilde{X} | Y$.

For example, suppose that $X \sim N(0, \Sigma)$, then i) becomes

$$(X, \tilde{X}) \sim \mathcal{N}(0, G), \quad \text{where} \quad G = \begin{bmatrix} \Sigma & \Sigma - \text{diag}\{s\} \\ \Sigma - \text{diag}\{s\} & \Sigma \end{bmatrix}.$$

Second-Order Knockoffs

- Instead of asking the swap property i), require the first two moments to be equivalent:

$$E(X) = E(\tilde{X})$$
$$cov(X, \tilde{X}) = \mathbf{G}, \text{ where } \mathbf{G} = \begin{pmatrix} \Sigma & \Sigma - \text{diag}\{s\} \\ \Sigma - \text{diag}\{s\} & \Sigma \end{pmatrix}$$

- Find s through convex optimization

Knockoff Procedure

- The statistics W_j's need to satisfy the **flip-sign property**: swapping the j-th variable with its knockoff has the effect of changing the sign of W_j.
- Run a model on $Y \sim (X, \widetilde{X})$,
  - e.g. least-squares linear regression, random forest, etc.
- Pick appropriate feature importance metric (denoted as Z_j)
  - e.g. absolute value of coefficients, Gini impurity
- Find an antisymmetric function between the feature importance metrics of the original and knockoff variables
  - e.g. $W_j = W(Z_j, \tilde{Z}_j) = Z_j - \tilde{Z}_j$

Knockoff Procedure

- Given nominal FDR q, e.g. $q = 0.10$,
- Calculate τ, a threshold for determining significance

$$\tau_q = \min\left\{ t > 0 : \frac{1 + |\{j : W_j \leq -t\}|}{|\{j : W_j \geq t\}|} \leq q \right\}$$

- Classify variable j as significant if $W_j > \tau_q$

Knockoff Procedure

- If variable is truly significant, importance metric should be **strongly different** between original and its knockoff
- Threshold τ is data-driven
- Alternative to p-value and confidence interval approaches
  - FDR can be properly controlled while taking the number of variables into account

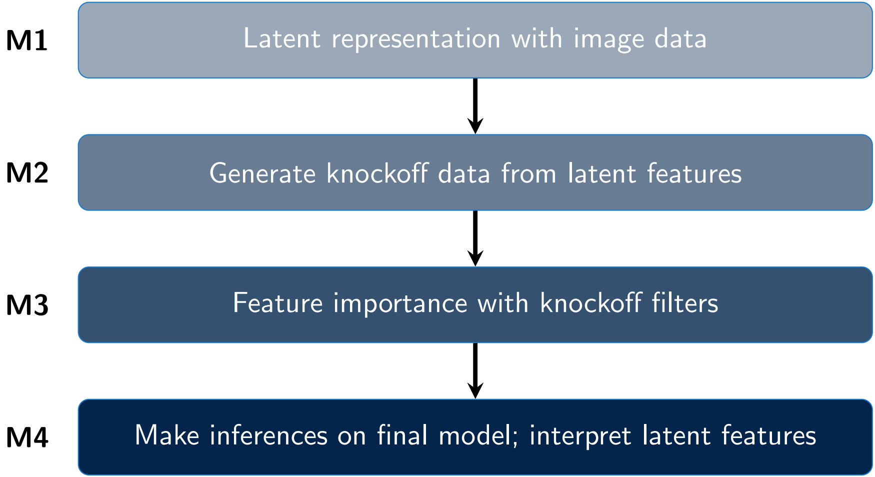
Overview

1. Motivations
2. Introduction
3. Proposed Methods
   - Notations
   - ImgKnock Procedure
4. Numerical Experiments
5. Summary

Notations

- $(\mathbf{X}, \mathbf{Y})$: original data (input and output) of size $n \times (p+1)$
- $\tilde{\mathbf{X}}$: knockoff data for $\mathbf{X}$ of size $n \times p$
- $(\mathbf{X}, \tilde{\mathbf{X}})$: combined data of size $n \times 2p$
- $\mathbf{Z}$, $\tilde{\mathbf{Z}}$: latent features (learned from original features)
- $\mathbf{S}$, $\tilde{\mathbf{S}}$: scores of feature importance
- $\mathbf{W_j}$: knockoff statistic derived from antisymmetric expression of S_j and $\tilde{S}_j$ for variable j

Framework


M1
Latent representation with image data
M2
Generate knockoff data from latent features
M3
Feature importance with knockoff filters
M4
Make inferences on final model; interpret latent features

Procedure with Latent Features

- Generate latent features from the original pixel features via a denoising autoencoder
  - **M1**: model to convert noisy/original features to latent features
- Initialize and create knockoff machines using latent features, which can then be used to generate knockoffs of latent features
  - **M2**: second-order and deep machines to generate knockoffs of latent features
- Use knockoff procedure to control FDR and run feature selection using algorithm of choice
  - **M3**: classification algorithm to run knockoff procedure and variable selection
    - ★ Logistic regression with LASSO penalty
    - ★ Use of absolute value of β coefficients for S
    - ★ $W_j = S_j - \tilde{S}_j$
- Use PCA and other dimension-reduction techniques to induce interpretation of latent features
  - **M4**: interpretation of latent features for understanding

M1 Generate Latent Features from Image Data

- Use of a denoising autoencoder (DAE)
- Add standard Gaussian noise to pixel values, with some scale factor $\eta \in [0, 1]$

$$X' = X + \eta\epsilon,$$

- Choose architecture for encoding and decoding, which contains mixture of convolution and normalization
  - Adapted from Tarun Kumar [4]
  - Narrow down to p' latent features
- Latent features can be used for knockoff generation

M2 Generate Knockoff Data from Latent Features

- Use of second-order and deep knockoff methodology
  - Second-order: semidefinite construction with tolerance of 10^{-5}
  - Deep: 40 epochs with epoch length of 20; 32-node width; $(\gamma, \lambda, \delta) = (1, 1, 1)$
- Second-order: Reliant on the first two moments being equivalent
- Deep: Reliant on various scoring metrics
- Feed latent features in to create knockoff machines

Illustration of M1 and M2

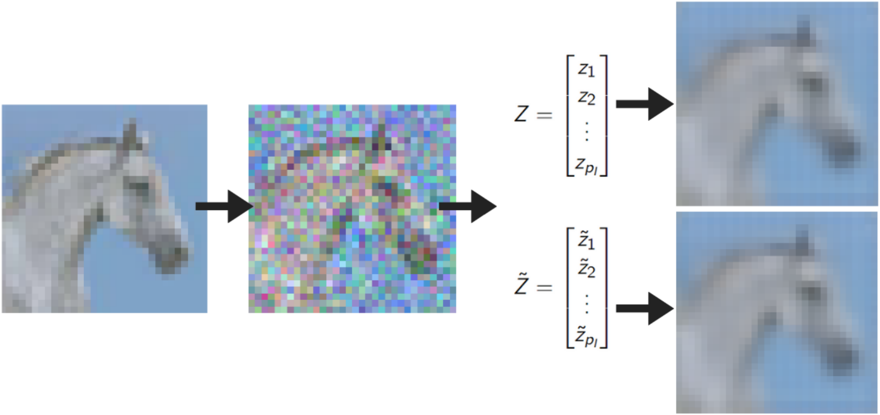


Figure: Illustration of M1 and M2

M3 Run Feature Selection Using Some Algorithm and Knockoff Filter

- Feed training data X into pre-created denoising models to get latent features X'
- Feed training latent features X' to get knockoff latent features $\tilde{X}$
- Use logistic regression with LASSO penalty on combined data $(X, \tilde{X})$
- Use of absolute value of β coefficients for S
- $W_j = S_j - \tilde{S}_j$

Feature Importance

- Use of logistic regression with various penalties
  - LASSO (L1), elastic, Ridge (L2)
  - $S_j = |\beta_j|$, $\tilde{S}_j = |\tilde{\beta}_j|$
- Random forest
  - $S_j =$ mean decrease in impurity for some variable j
- Boosting
  - $S_j =$ weight-based importance for variable j

Knockoff Filter

- Knockoff statistic is still calculated as difference of scores

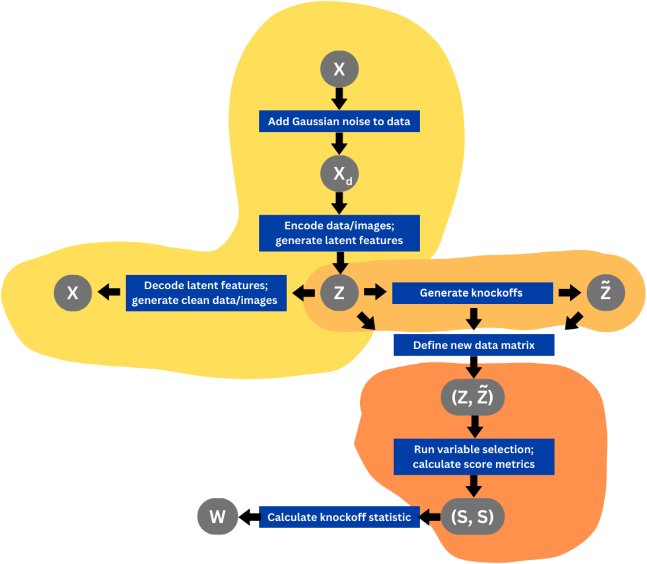
$$W_j = S_j - \tilde{S}_j$$

- Remaining procedures are the same
- Interpretation
  - Use of PCA to identify strongest features
  - Can decode features to generate encoded images

M4 Make Inferences on Final Model; Interpret Latent Features

- Interpret latent features and map to clinical features
- Use of PCA on training latent features and knockoff latent features
- Use decoder aspect of autoencoder to regenerate images from latent features

Latent Representation Learning



Overview

1. Motivations
2. Introduction
3. Proposed Methods
4. Numerical Experiments
   - MNIST Data
   - Simulations
   - CIFAR-10 Data
   - Glacuoma Data
5. Summary

MNIST Data

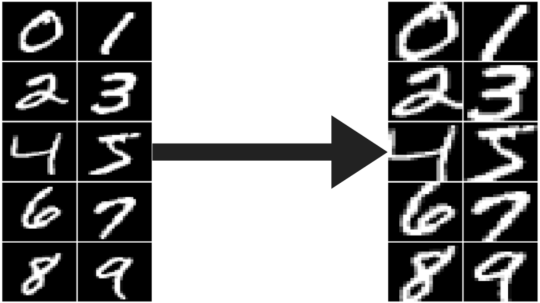
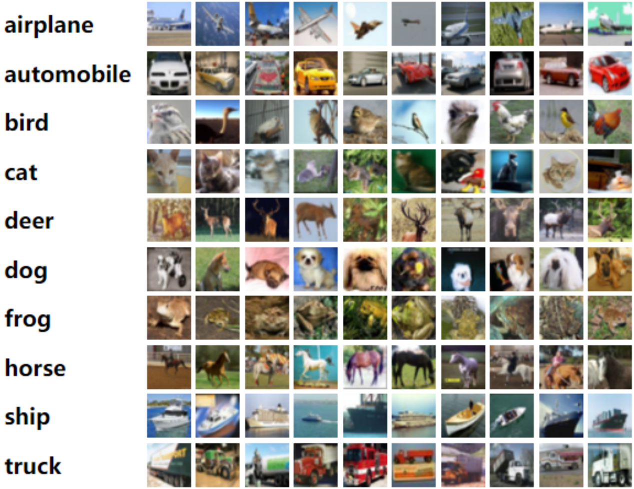


Figure: Conversion of MNIST Data from 28x28 ($p = 784$) to 20x20 ($p = 400$)

- Original X_i's are 20×20 grey scale images.
- Z_i's are latent feature vectors learned from Denoising Autoencoder (DAE), of length 32.
- Sample size $N = 50,000$ for multi-class classification
- Sample size $N = 10,000$ for two-class classification

CIFAR10

- Original X_i's are $32 \times 32 \times 3$ color images;
- Latent Z_i's are feature vectors of length 128, learned from DAE.



Simulated Truth

- Simulate latent features Z_i's from the empirical distributions learned from the MNIST or CIFAR10 data. $Z_i \sim \widehat{F}_{Z,\text{MNIST}}$ or $Z_i \sim \widehat{F}_{Z,\text{CIFAR10}}$
- Simulate true class labels following

$$p_i = \frac{\exp(Z_i^{\mathrm{T}} \boldsymbol{\beta}^0)}{1 + \exp(Z_i^{\mathrm{T}} \boldsymbol{\beta}^0)};$$
$$Y_i = Binom(1, p_i),$$

  where $\boldsymbol{\beta}^0$ is a fixed realization of a sparse vector. For example, $S_0 = \{j : \beta_j^0 \neq 0\}$, and $\beta_j^0 = \pm 1$ for $j \in S_0$.
- Apply ImgKnock to the simulated datasets and evaluate the empirical performances in terms of TP, FP, FDR, etc.

Simulation 1 MNIST Data

- $Z_i \in R^q$, $q = 32$, $N = 1000$;
- Z_i's are i.i.d. bootstrap samples of the empirical latent features learned from the MNIST data;
- β^0 with $S_0 = \{4, 8, 12, \ldots, 32\}$ and 4 $+1$'s and 4 -1's.

	q	0.050	0.100	0.200	0.300	0.400	0.500	0.750
2nd	TPR	0.000	0.000	0.380	0.620	0.680	0.860	0.880
	FPR	0.000	0.000	0.027	0.173	0.255	0.391	0.482
	FDR	0.000	0.000	0.073	0.247	0.342	0.480	0.543
Deep	TPR	0.000	0.000	0.400	0.840	0.920	1.000	1.000
	FPR	0.000	0.000	0.055	0.200	0.300	0.482	0.609
	FDR	0.000	0.000	0.129	0.272	0.318	0.476	0.569

Simulation 1 MNIST Data

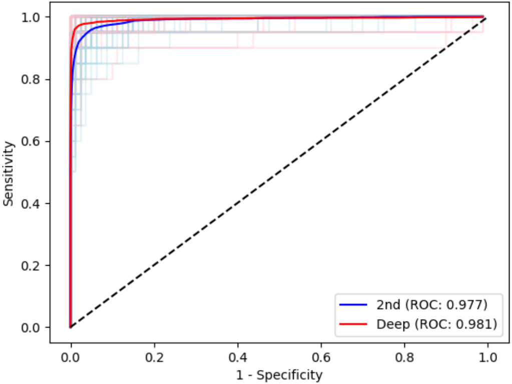


Figure: ROC Curve for Situation #1

Simulation 1 MNIST Data

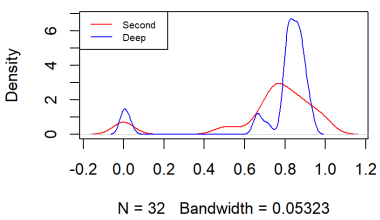


Figure: Left: $Cor(X_j, \tilde{X}_j)$; Right: $Cor(X_1, \tilde{X}_j)$.

Simulation 2 CIFAR-10

- $Z_i \in R^q$, $q = 128$, $N = 1000$;
- Z_i's are i.i.d. samples of the multivariate normal distribution with mean and covariance from the empirical latent features learned from the CIFAR-10 data;
- β^0 with $|S_0| = 20$, 10 $+1$'s and 10 -1's.

Simulation 2 CIFAR-10

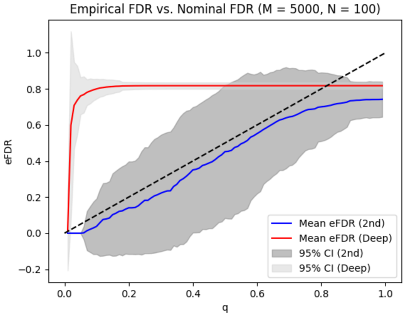


Figure: Average FDR vs. Nominal FDR for Simulation 2 with CIFAR-10 Data.

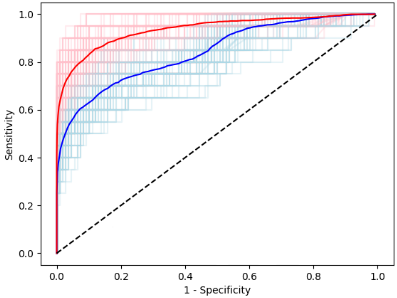


Figure: ROC Curve for Simulation 2 with CIFAR-10 Data.

Simulation 2 CIFAR-10

Figure: Distribution of Latent Means of 128 Features between two classes

Simulation 2 CIFAR-10

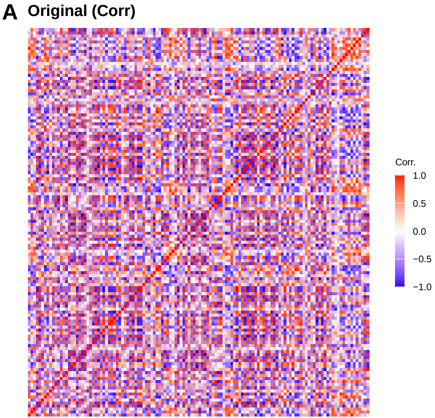


Figure: PCA of 128 Features between two classes

Overview

1. Motivations
2. Introduction
3. Proposed Methods
4. Numerical Experiments
5. Summary
   - Next Steps

Summary

- We proposed ImgKnock, a novel framework for knockoff inference with image data, via latent representation learning.
- Our method can make inference with the latent endogenous features of the images.
- Knockoff feature generation and selection guarantees FDR control.

Next Steps:

- Application to the fundus images for glaucoma detection.
- Interpretation of latent features with glaucoma data.

Advantages of Deep Learning in Ophthalmology

- **Scalability**: Deep learning models can be deployed across multiple clinics and hospitals, providing scalable solutions for glaucoma screening.
- **Efficiency**: Rapid processing of large volumes of images, aiding in faster diagnosis and management.
- **Integration with Clinical Practice**: Deep learning tools can be integrated with electronic health records (EHR) systems for seamless workflow.

References

R. F. Barber and E. J. Candès.
Controlling the false discovery rate via knockoffs.
2015.

R. F. Barber and E. J. Candès.
A knockoff filter for high-dimensional selective inference.
2019.

E. Candes, Y. Fan, L. Janson, and J. Lv.
Panning for gold:‘model-x’knockoffs for high dimensional controlled variable selection.
Journal of the Royal Statistical Society Series B: Statistical Methodology, 80(3):551–577, 2018.

T. K.
Autoencoder denoising image - mnist, cifar10, 2020.
Accessed: 2024-07-08.

A. Krizhevsky, G. Hinton, et al.
Learning multiple layers of features from tiny images.
2009.

Y. Romano, M. Sesia, and E. Candès.
Deep knockoffs.
Journal of the American Statistical Association, 115(532):1861–1872, 2020.

Deep Knockoffs

- Knockoff copy returned based on random mapping f_θ and noise vector V

$$\tilde{X} = f_\theta(X, V)$$

- Scoring function J evaluates empirical distribution of $(X, \tilde{X})$,

$$J(X, \tilde{X}) = \gamma J_{MMD}(X, \tilde{X}) + \lambda J_{\text{second-order}}(X, \tilde{X}) + \delta J_{\text{decorrelation}}(X, \tilde{X})$$

  - J_{MMD}, maximum mean discrepancy;
  - $J_{\text{second-order}}$, matching of the second moments of X and $\tilde{X}$;
  - $J_{\text{decorrelation}}$, penalize large pairwise empirical correlations between X and $\tilde{X}$.
- Fit deep neural networks to obtain the knockoff machine $\widehat{f_\theta}$.

Knockoff Procedure - Deep

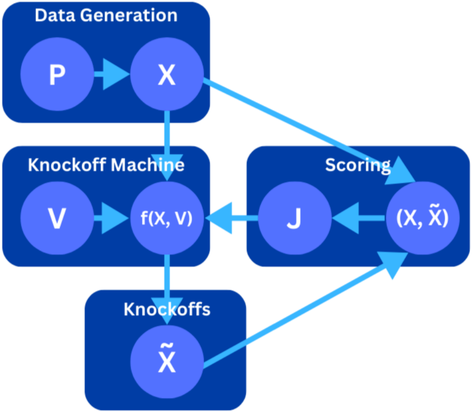


Figure: Structure of Deep Knockoffs [6]