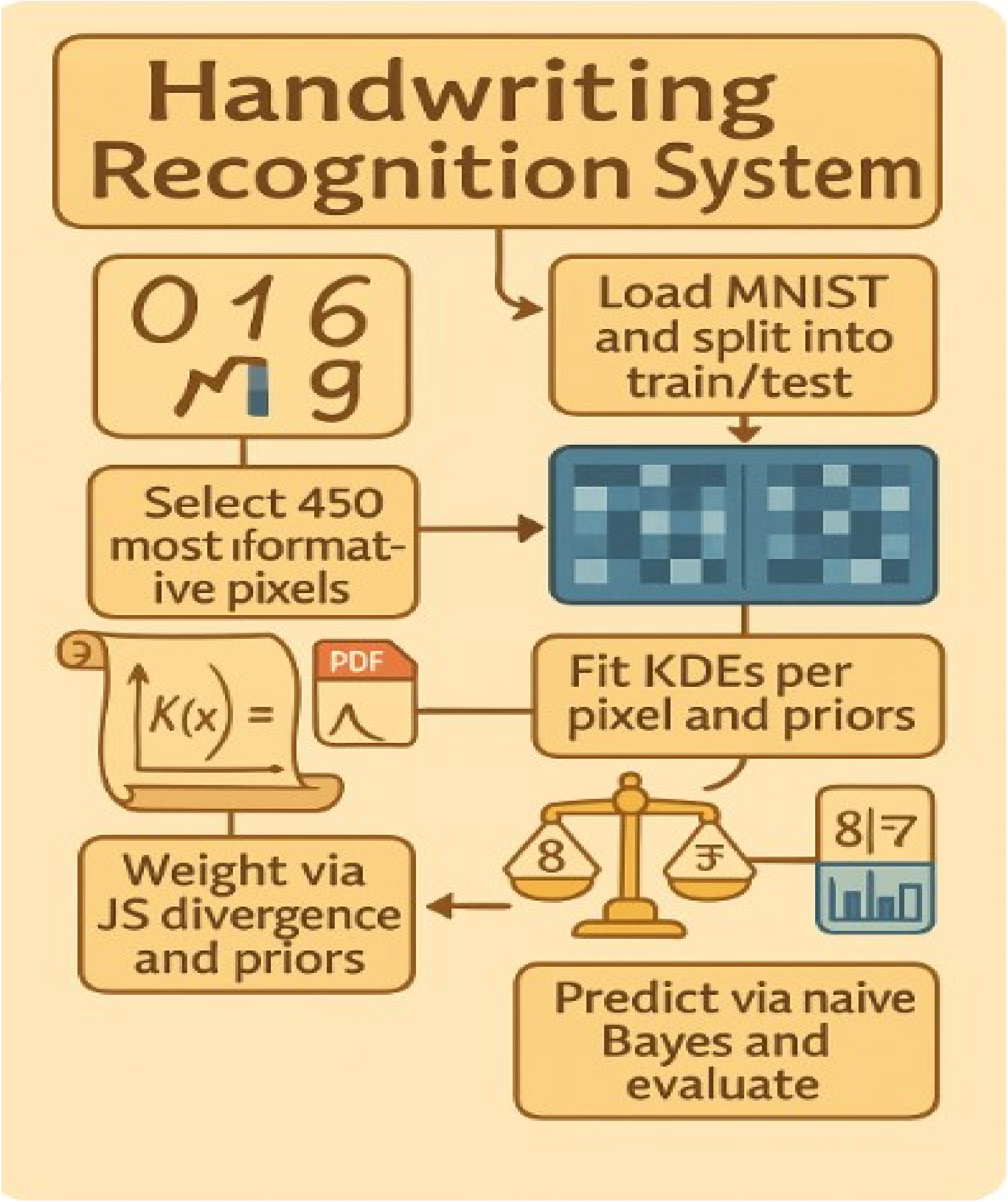
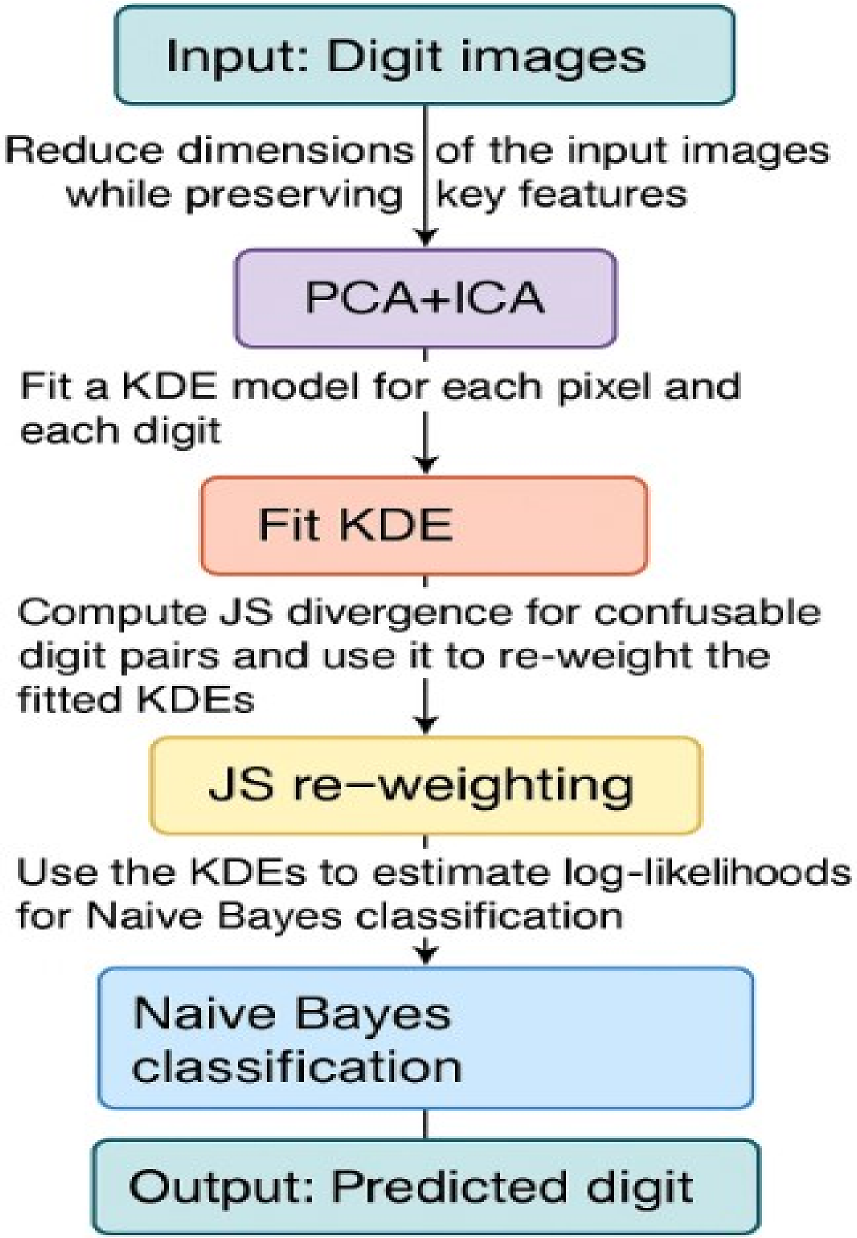
KDE-Based Digit Classifier



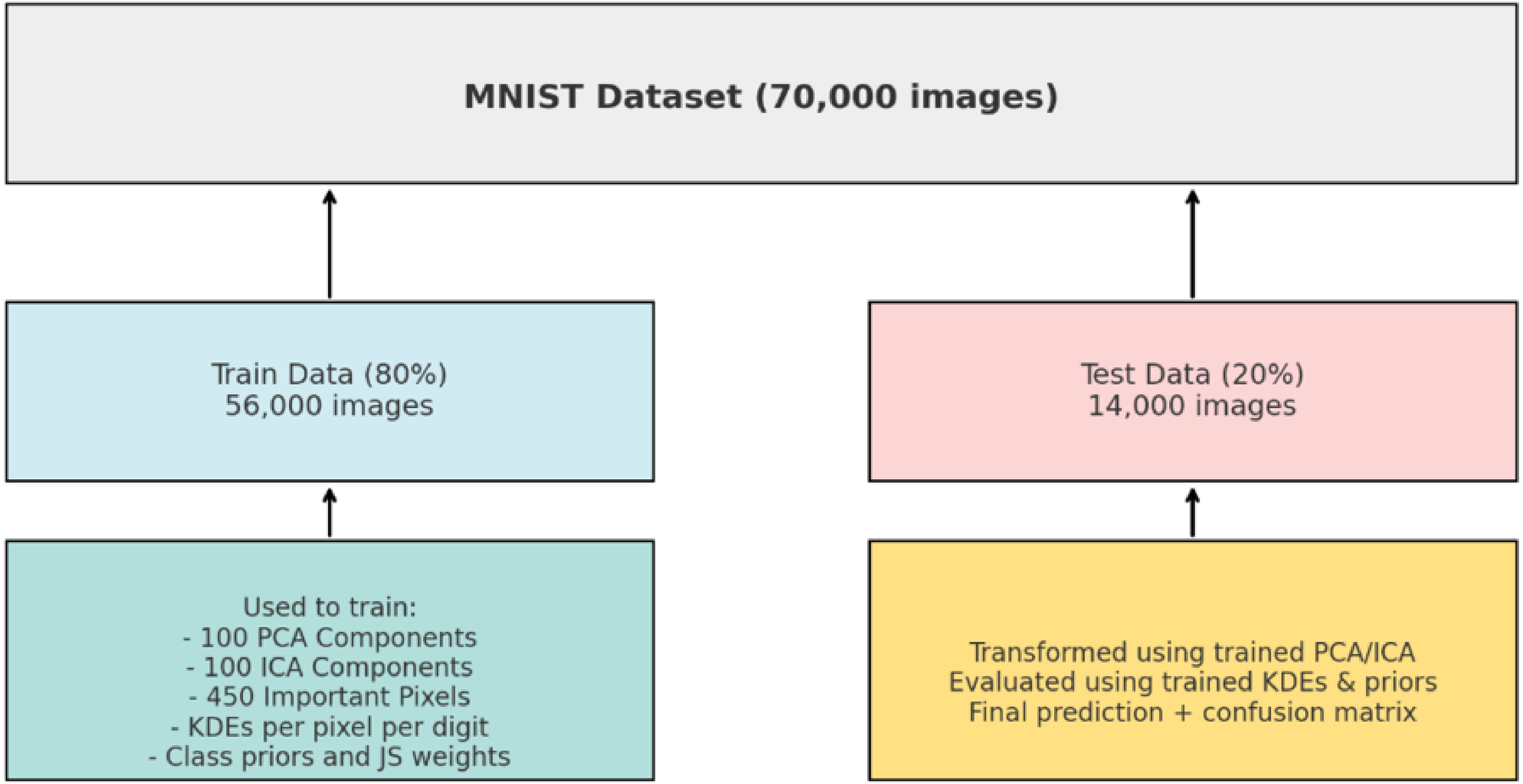
Theory Foundation

- Naive Bayes is a **probabilistic classification algorithm** based on **Bayes' theorem**, which assumes that features are **conditionally independent** given the class.
- We apply this principle to our image classification system:
  - Each **pixel position** is treated as an independent feature.
  - We assume that **pixel values at different positions** do not influence each other.
- This allows us to **model the likelihood of each digit** by estimating **per-pixel probability densities** using **Kernel Density Estimation (KDE)**.
- Classification then becomes a product of per-pixel probabilities.

System Overview



Data Usage Flow in KDE-Based Prediction System

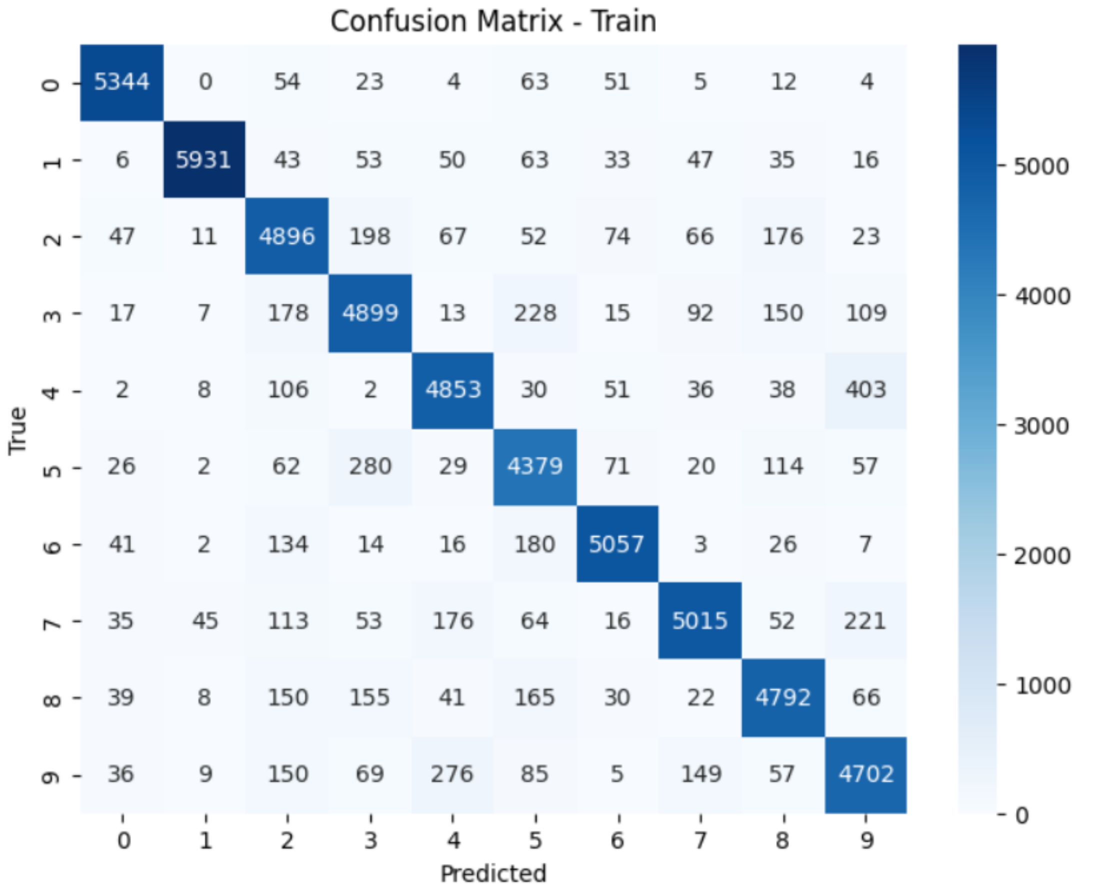


System Performance and Runtime Summary

- **Train Accuracy: 0.8905**
- **Test Accuracy: 0.8719**

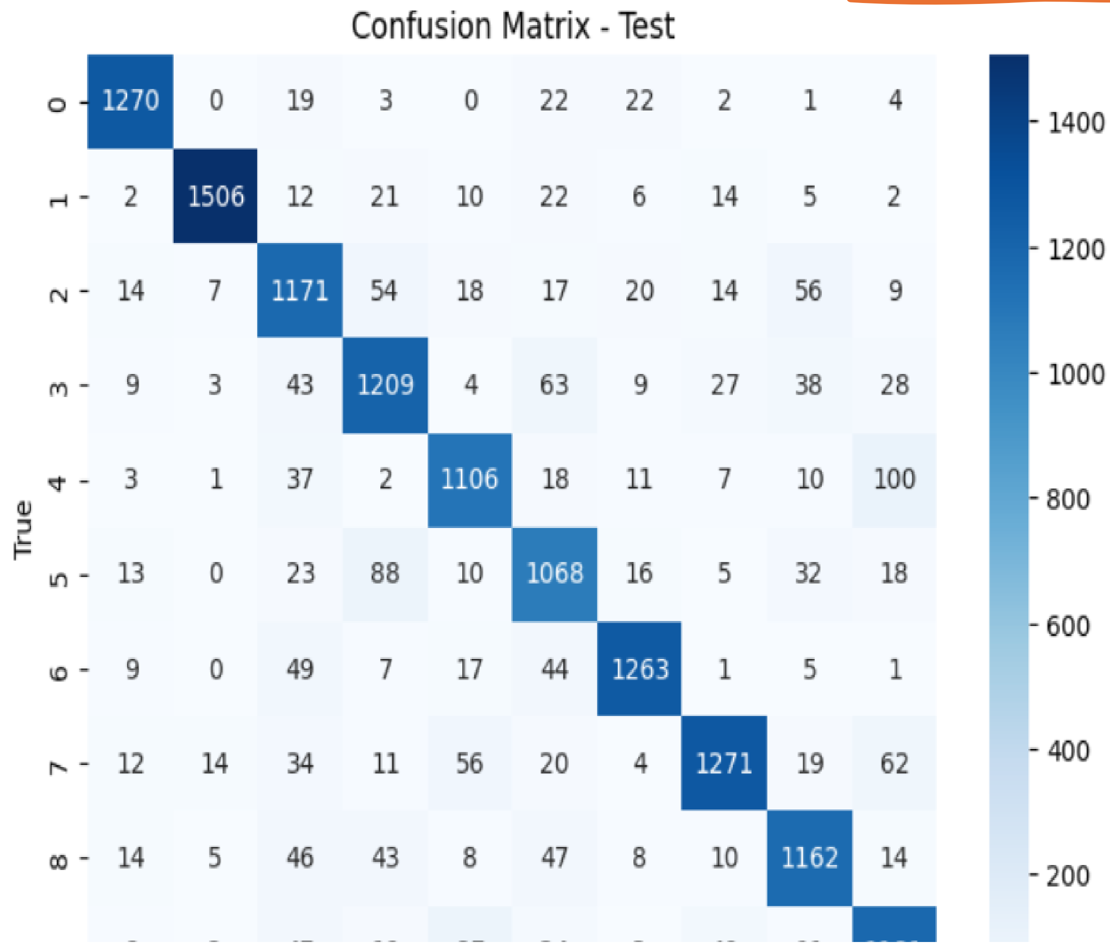
Phase	Sub-step	Time (s)	Description
Data Prep	Load & Split	`9.46`	Load MNIST from OpenML, normalize pixel values, and split into train/test sets
	Variance Filtering	`0.88`	Remove pixels with variance below 0.005 to reduce noise
	PCA + ICA	`88.07`	Apply PCA and ICA to reduce dimensionality while preserving key features
	Mutual Info Selection	`271.84`	Select 450 pixels with highest mutual information relative to class labels
Model Fitting	KDE Fitting	`149.69`	Fit a PDF per digit (0–9) for each selected pixel using kernel density estimate
Evaluation	Train Prediction	`402.79`	Predict and evaluate 56,000 training images using log-likelihood summation
	Test Prediction	`100.70`	Predict and evaluate 14,000 test images
TOTAL	—	**1023.43**	Complete end-to-end runtime: ~17 minutes

Classification Report (Train Set)



	precision	recall	f1-score	support
0	0.96	0.96	0.96	5560
1	0.98	0.94	0.96	6277
2	0.83	0.87	0.85	5610
3	0.85	0.86	0.86	5708
4	0.88	0.88	0.88	5529
5	0.82	0.87	0.85	5040
6	0.94	0.92	0.93	5480
7	0.92	0.87	0.89	5790
8	0.88	0.88	0.88	5468
9	0.84	0.85	0.84	5538
accuracy			0.89	56000

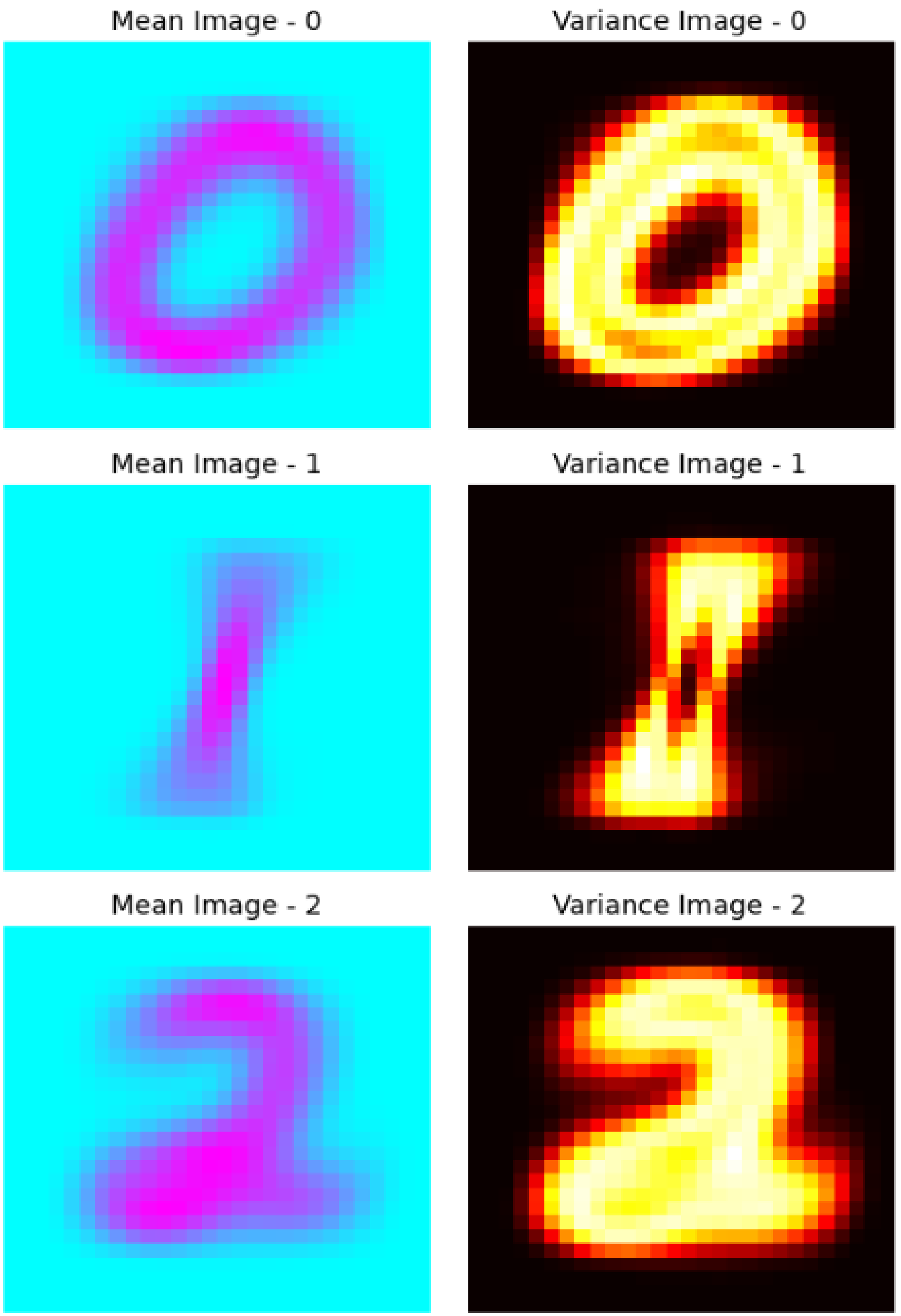
Classification Report (Test Set)



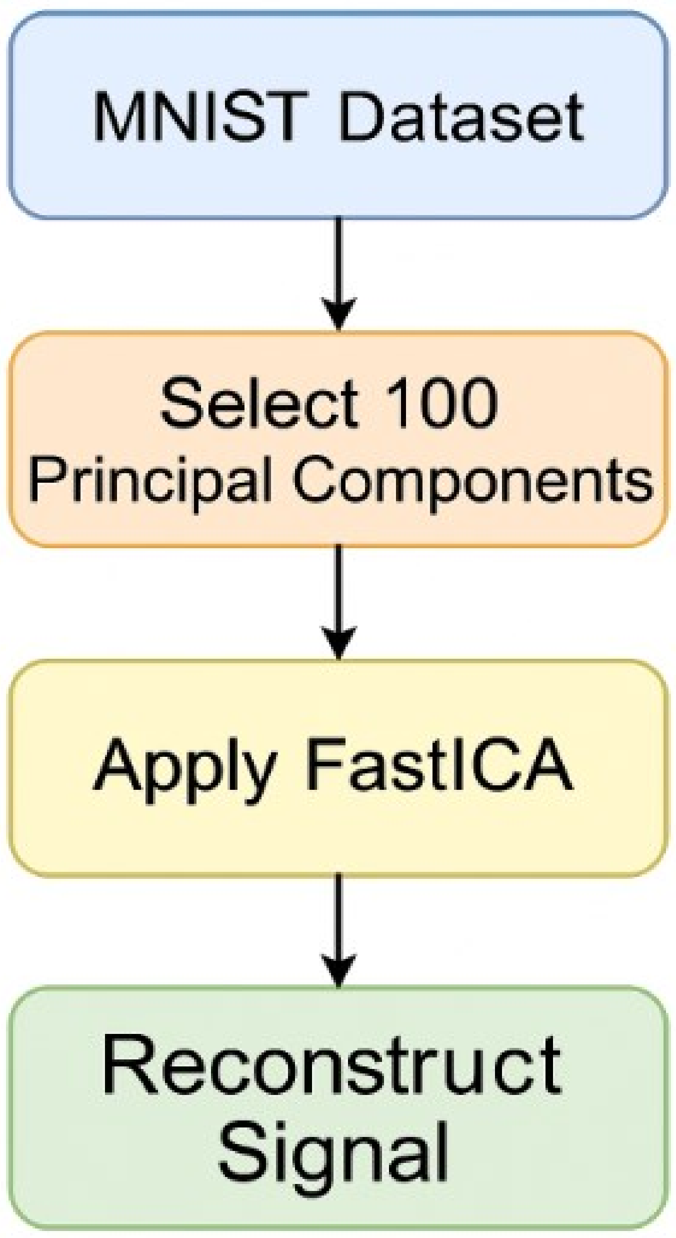
	precision	recall	f1-score	support
0	0.94	0.95	0.94	1343
1	0.98	0.94	0.96	1600
2	0.79	0.85	0.82	1380
3	0.83	0.84	0.84	1433
4	0.84	0.85	0.85	1295
5	0.79	0.84	0.82	1273
6	0.93	0.90	0.92	1396
7	0.91	0.85	0.88	1503
8	0.87	0.86	0.86	1357
9	0.83	0.83	0.83	1420
accuracy			0.87	14000

Digit Mean and Variance Heatmaps

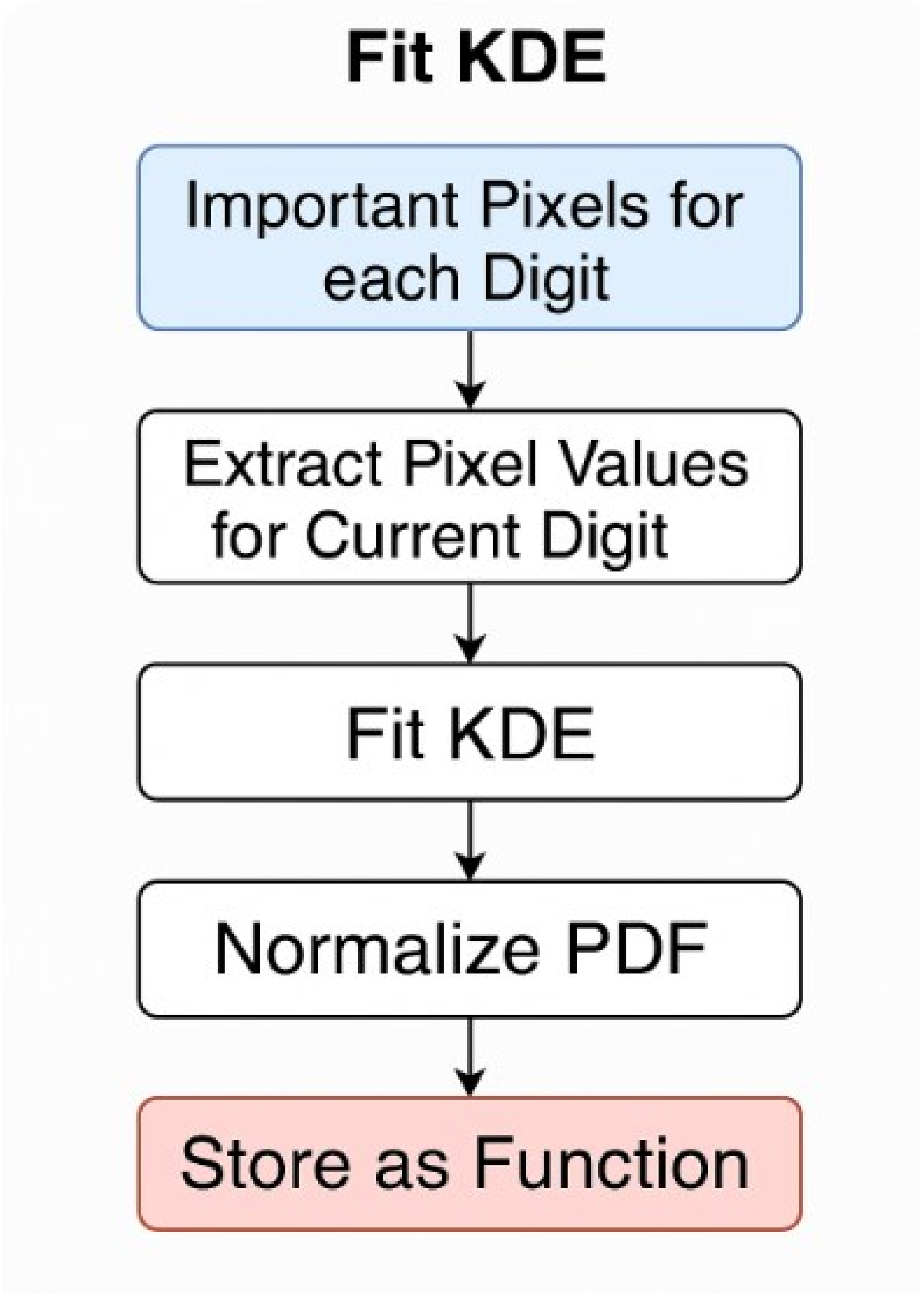
- **Mean images** show the typical shape and structure of each digit.
- **Variance images** highlight unstable pixel regions — where people write differently.
- Heatmaps help identify **informative pixels** for KDE fitting and enhance **digit separability** in classification.



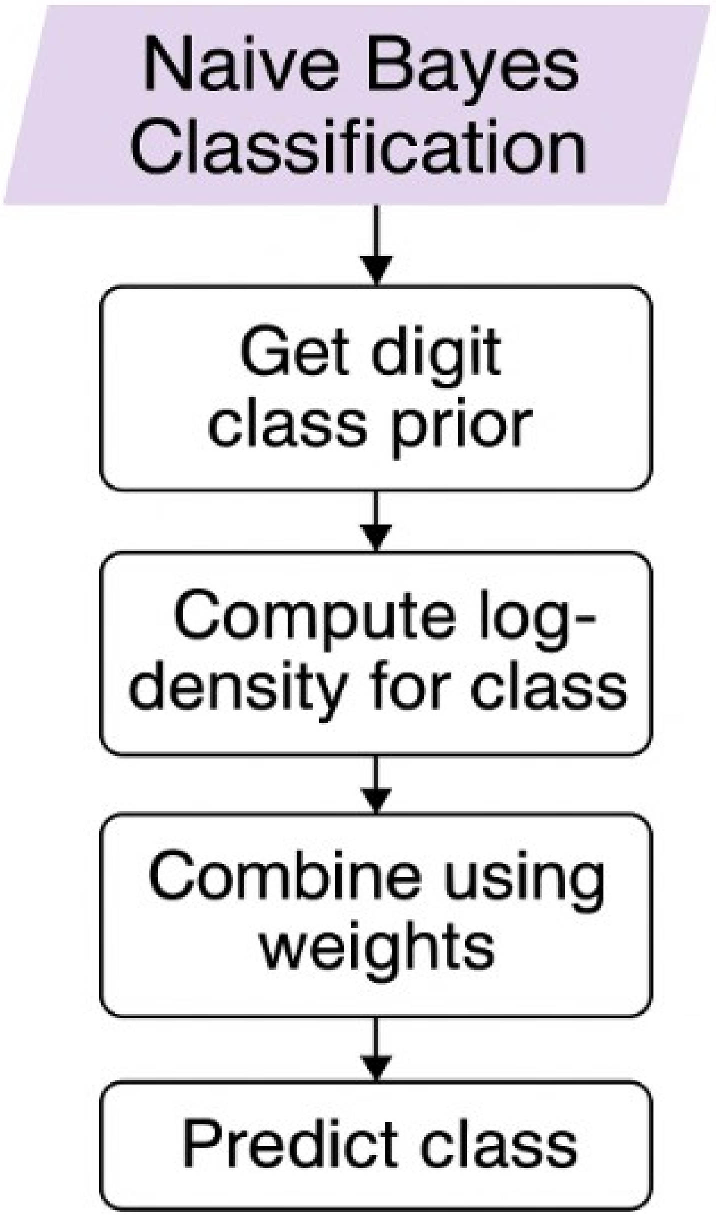
Preprocessing: Dimensionality Reduction



Fitting KDE Models per Digit and Pixel



Naive Bayes Classification Process



Understanding the KDE Curve

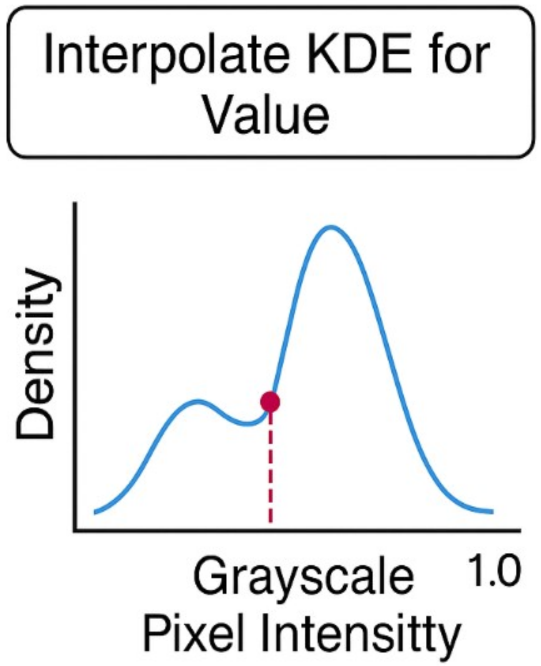
What is the KDE Curve?

The X-axis is the grayscale pixel intensity, from black (0) to white (1).

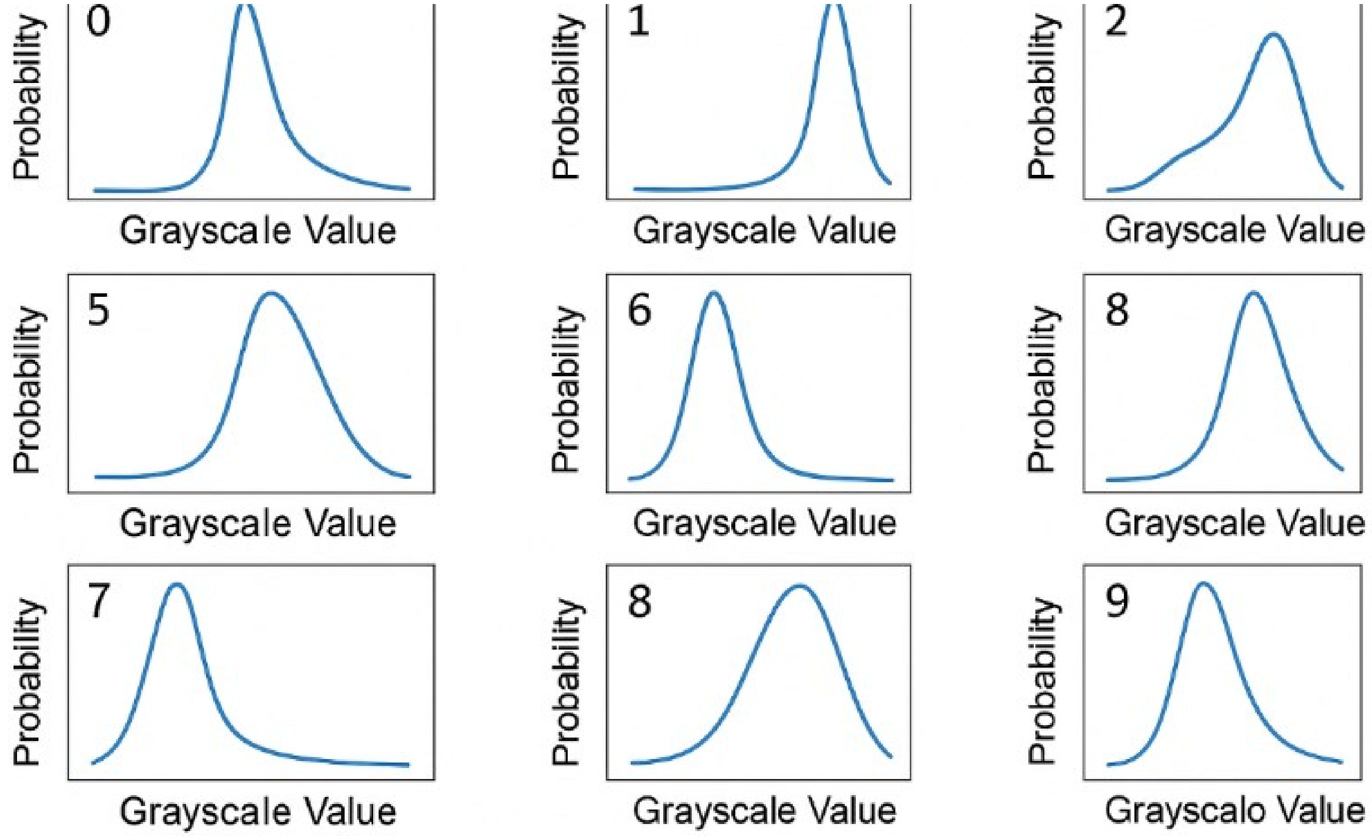
The Y-axis shows **probability density**, indicating how common each pixel value is for the class.

In simple terms:

The curve represents how the pixel typically looks for the class.



PDF Distributions at Pixel Position (10, 15)



From density function for position (10, 15)

Grayscale value: Pixel insensity

iles

Analyse your files with code written by Gemini

Upload

..

pdf_0_0_digit3.npz

pdf_0_0_digit4.npz

pdf_0_0_digit5.npz

pdf_0_0_digit6.npz

pdf_0_0_digit7.npz

pdf_0_0_digit8.npz

pdf_0_0_digit9.npz

pdf_0_1_digit0.npz

pdf_0_1_digit1.npz

pdf_0_1_digit2.npz

pdf_0_1_digit3.npz

pdf_0_1_digit4.npz

pdf_0_1_digit5.npz

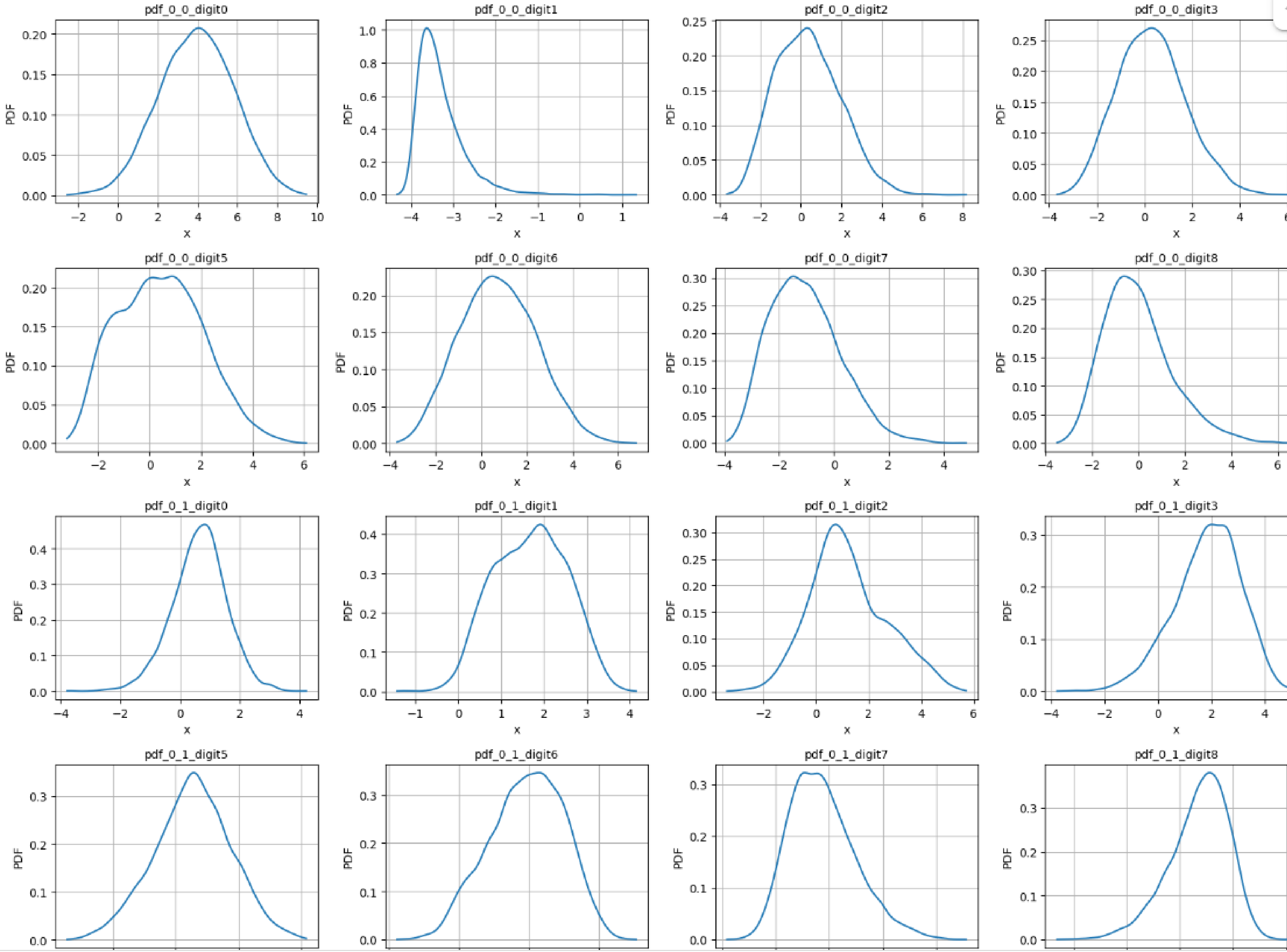
pdf_0_1_digit6.npz

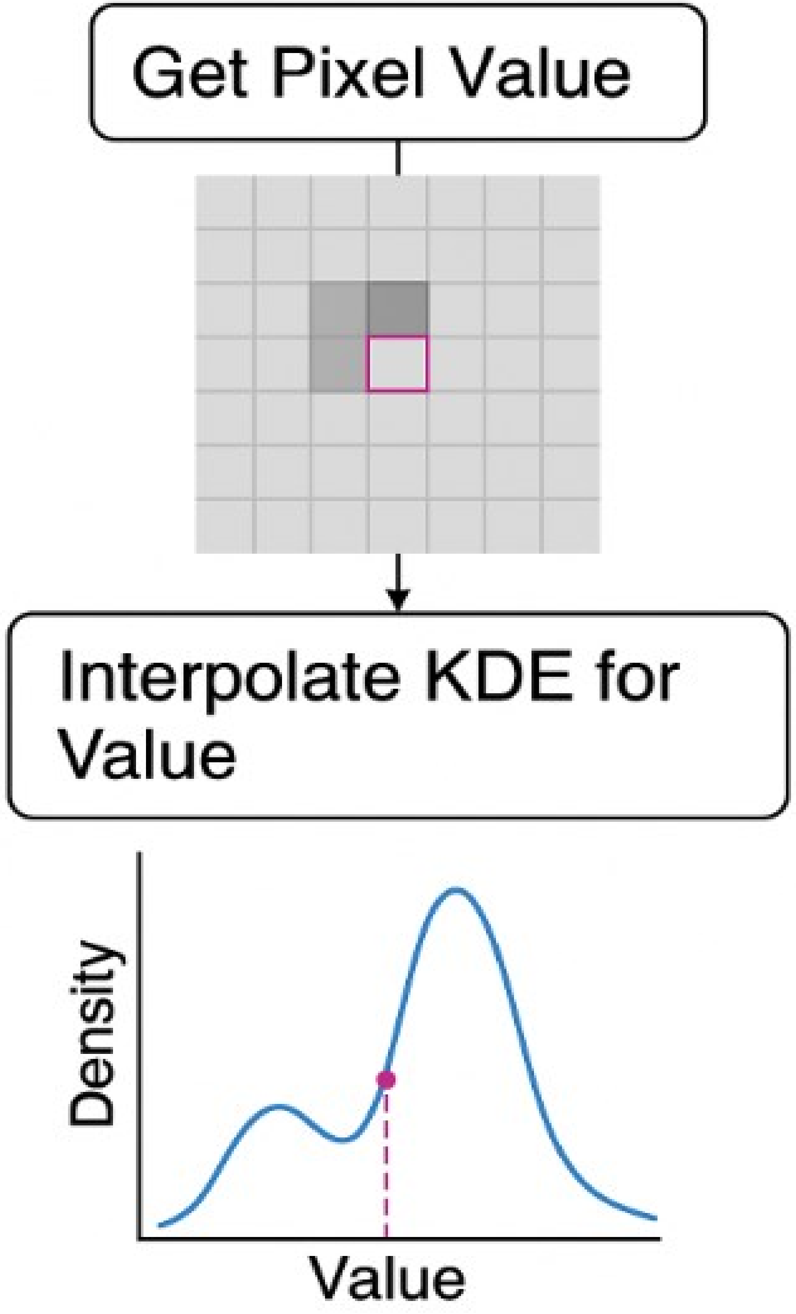
pdf_0_1_digit7.npz

pdf_0_1_digit8.npz

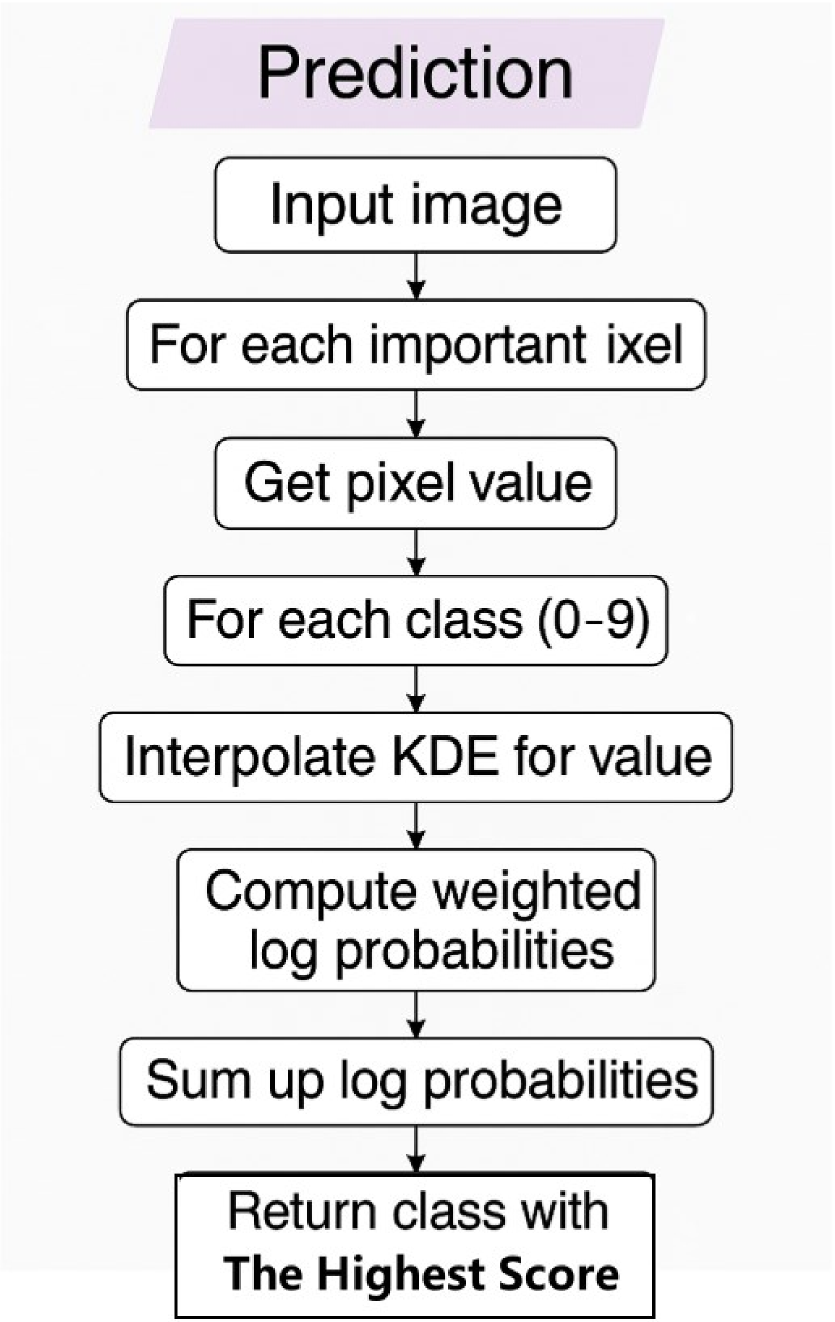
pdf_0_1_digit9.npz

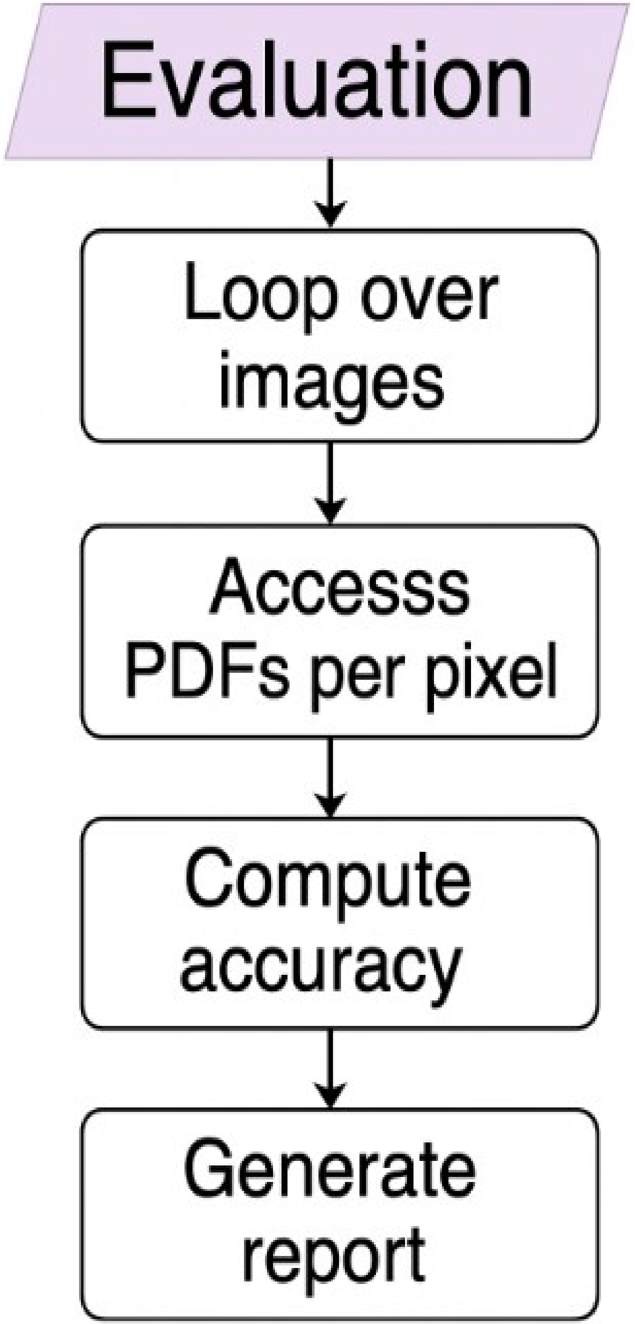
isk 70.77 GB available



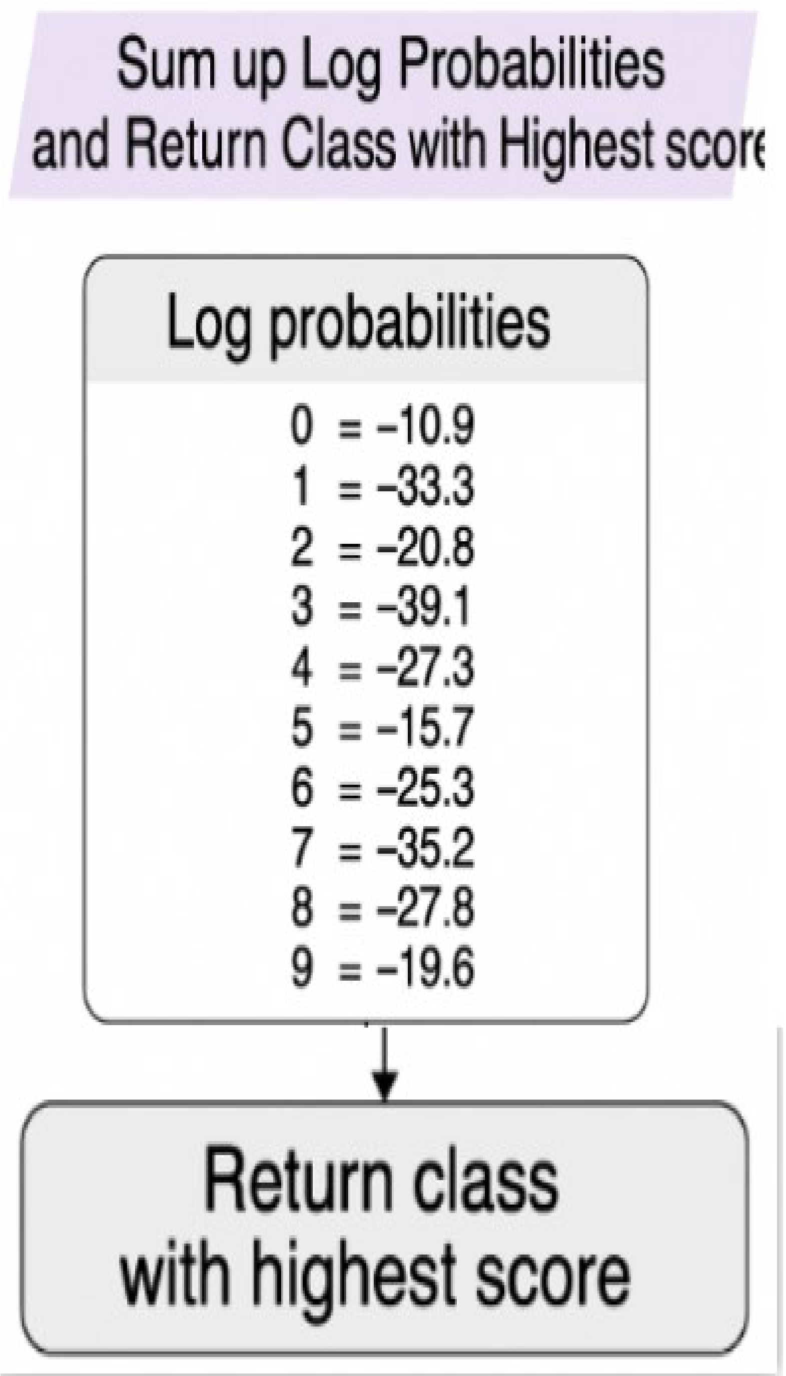

Get Pixel Value
Interpolate KDE for Value
Density
Value

Prediction


Prediction
Input image
For each important ixel
Get pixel value
For each class (0-9)
Interpolate KDE for value
Compute weighted log probabilities
Sum up log probabilities
Return class with
The Highest Score


Evaluation
Loop over images
Accesss PDFs per pixel
Compute accuracy
Generate report

Evaluation


Sum up Log Probabilities and Return Class with Highest score
Log probabilities
0 = −10.9
1 = −33.3
2 = −20.8
3 = −39.1
4 = −27.3
5 = −15.7
6 = −25.3
7 = −35.2
8 = −27.8
9 = −19.6
Return class with highest score