Program Sets

> Updated-2024-4-23

---

1. SNAKE: Shape-aware Neural 3D Keypoint Field

> 3D Scene Reconstruction

🥛 Accepted by NIPS. [Click Here For paper Link]

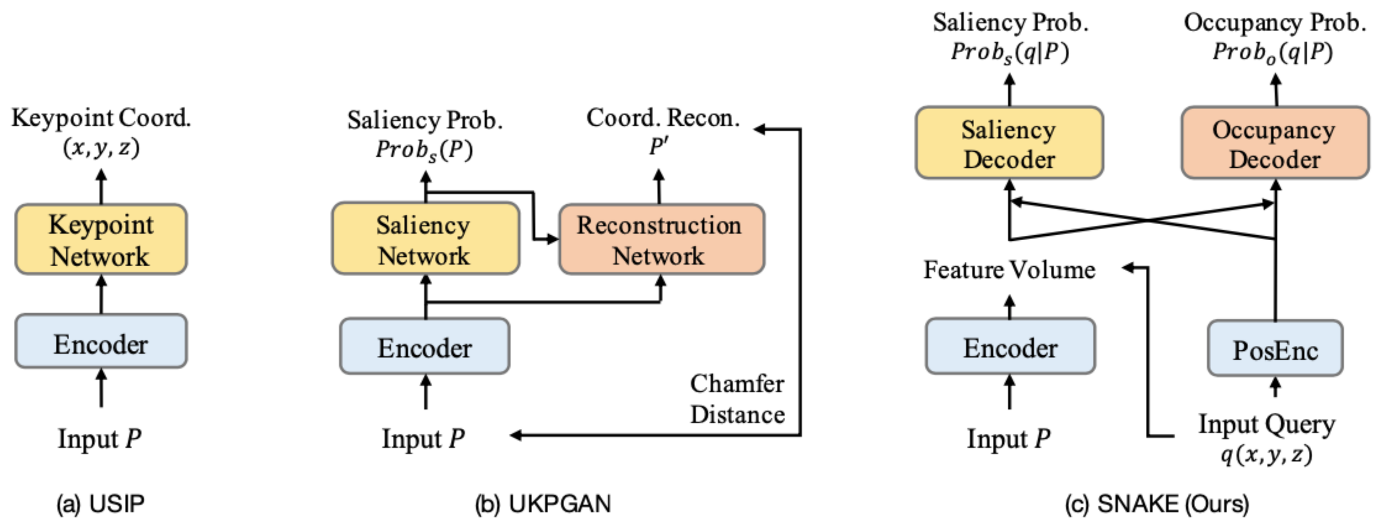
1.1 Situation:

Existing methods either seek salient features according to statistics of different orders or learn to predict keypoints that are invariant to transformation. Nevertheless, the idea of incorporating shape reconstruction into 3D keypoint detection is under-explored.

1.2 Target

New problem formulation: shape reconstruction + keypoint detection(supervise each other's training)

1.3 Action:

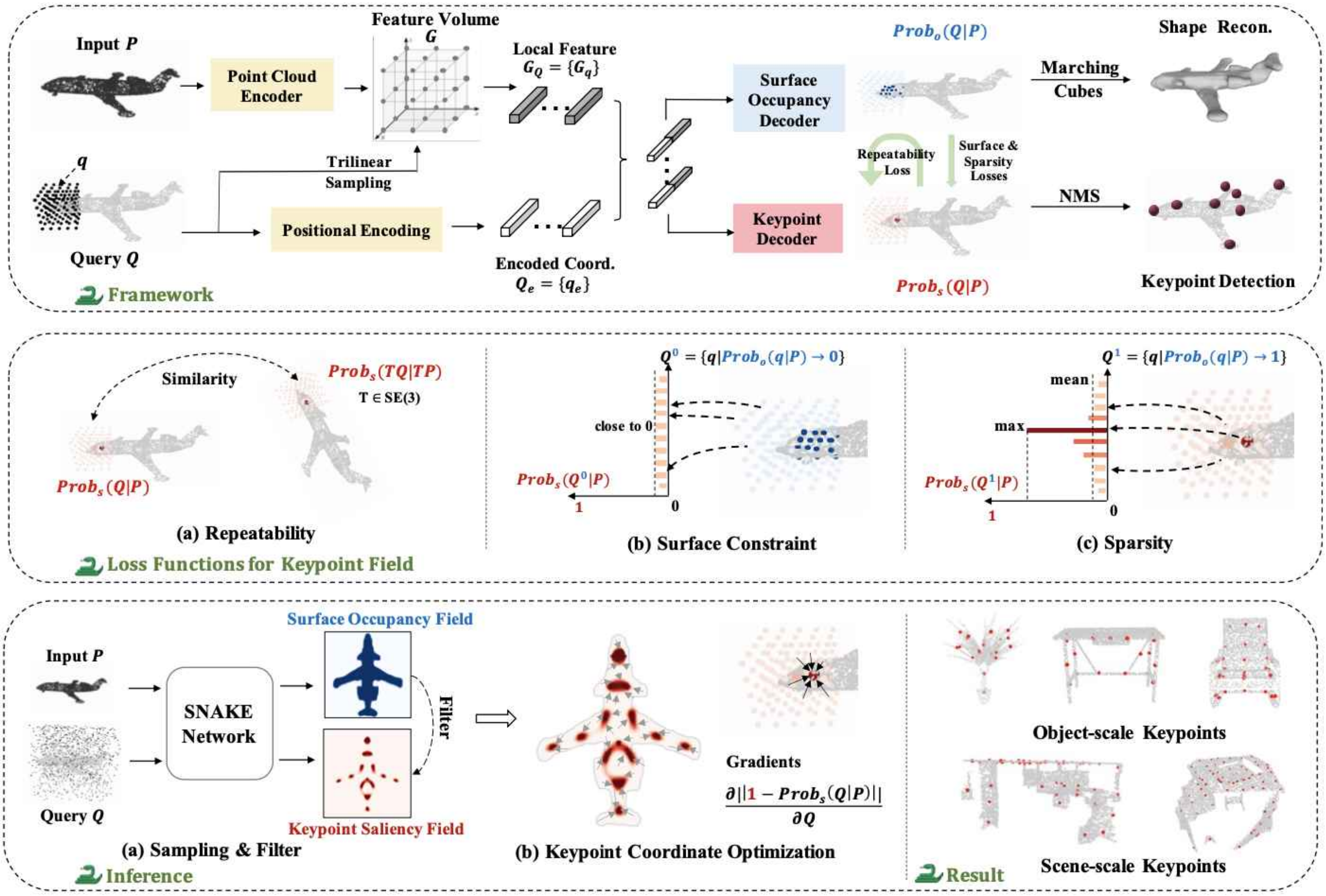


our proposed framework:shape reconstruction + keypoint detection(supervise each other's training)

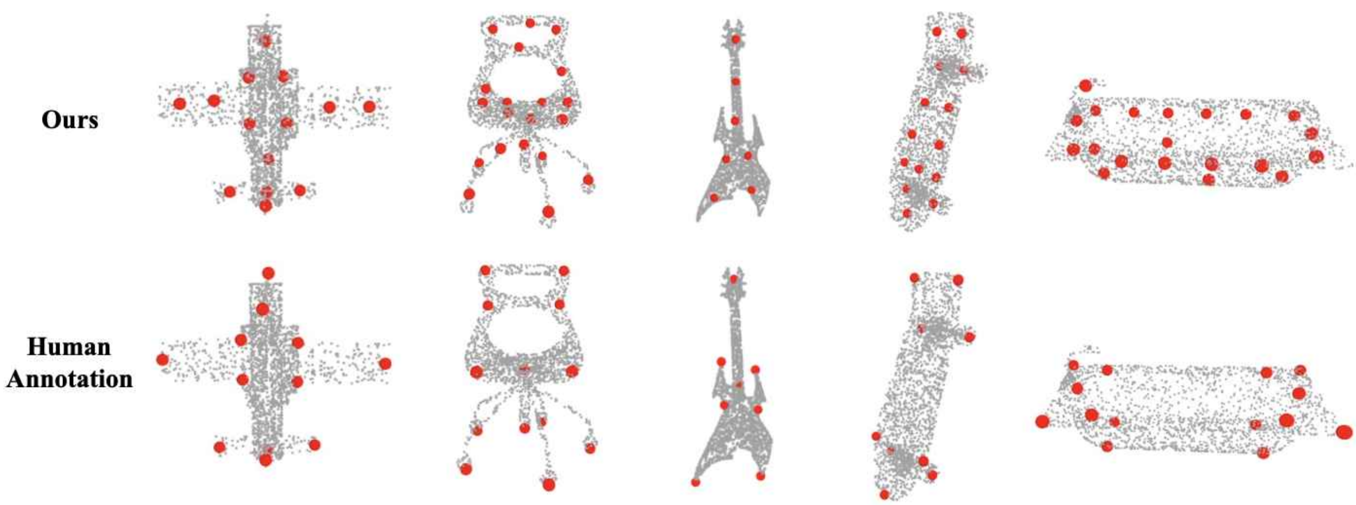
Repeatability Loss -- Learn Pose Invariant

Surface Constraint Loss -- Constraint Keypoint on the Surface

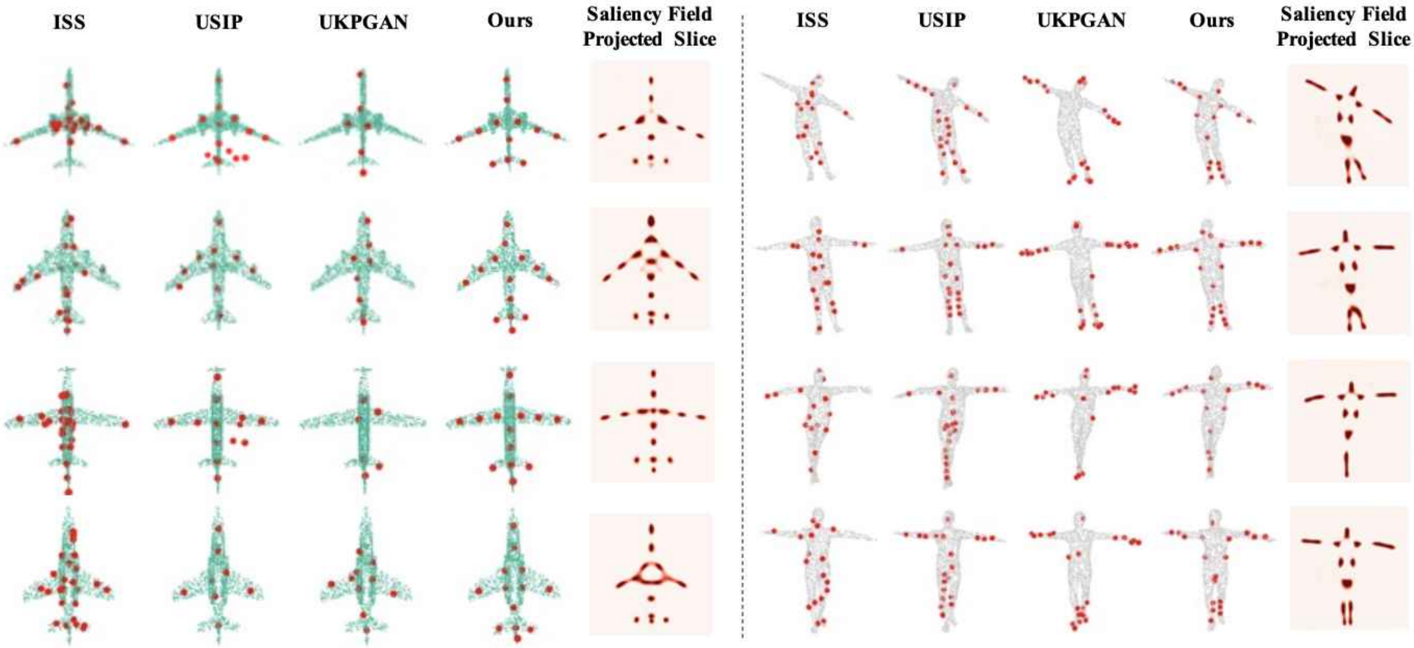
Sparsity Loss -- Avoid All zero Saliency Function



1.4 Result:



SNAKE generates 3D keypoints consistent with human semantic annotation, even without such supervision



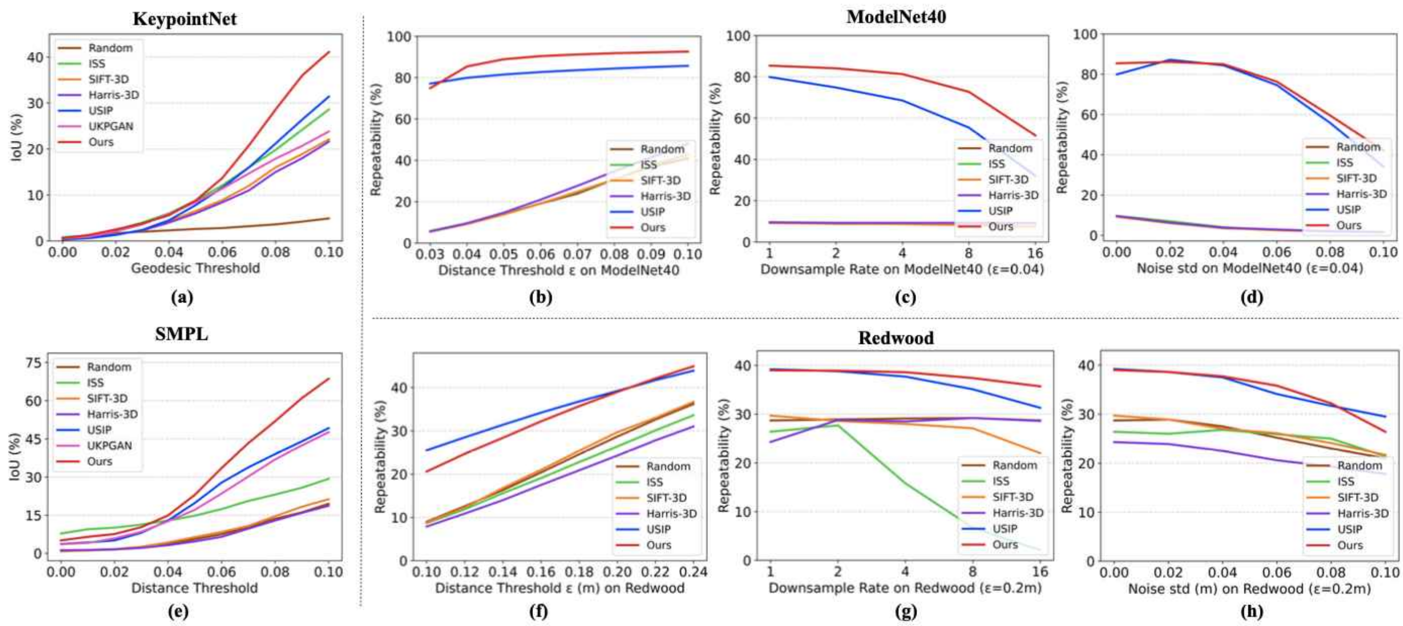


Figure 5: Quantitative results on four datasets. Keypoint semantic consistency (a)(e) on KeypointNet and SMPL. Relative repeatability for two-view point clouds with different distance threshold (b), downsample rate (c), Gaussian noise $\mathcal{N}(0, \sigma_{noise})$ (d) on ModelNet40. The results of (f)(g)(h) are tested on Redwood with the same settings in (b)(c)(d). The specific numerical results can be found in the Appendix.

SNAKE outperforms counterparts in terms of repeatability, especially when the input point clouds are down-sampled.

2. An Adapter for Interactive Object Retrieval on the Shelf

Robot Manilupation

🎼 Lots of potential use cases: Library/Mall/etc.

2.1 Situation:

The storage space on the shelf is usually limited and items are stacked next to each other, which brings great challenges for the robot to retrieve a target object from the shelf.

2.2 Target

Make Robot Arm able to grasp objects by interacting with the environment.

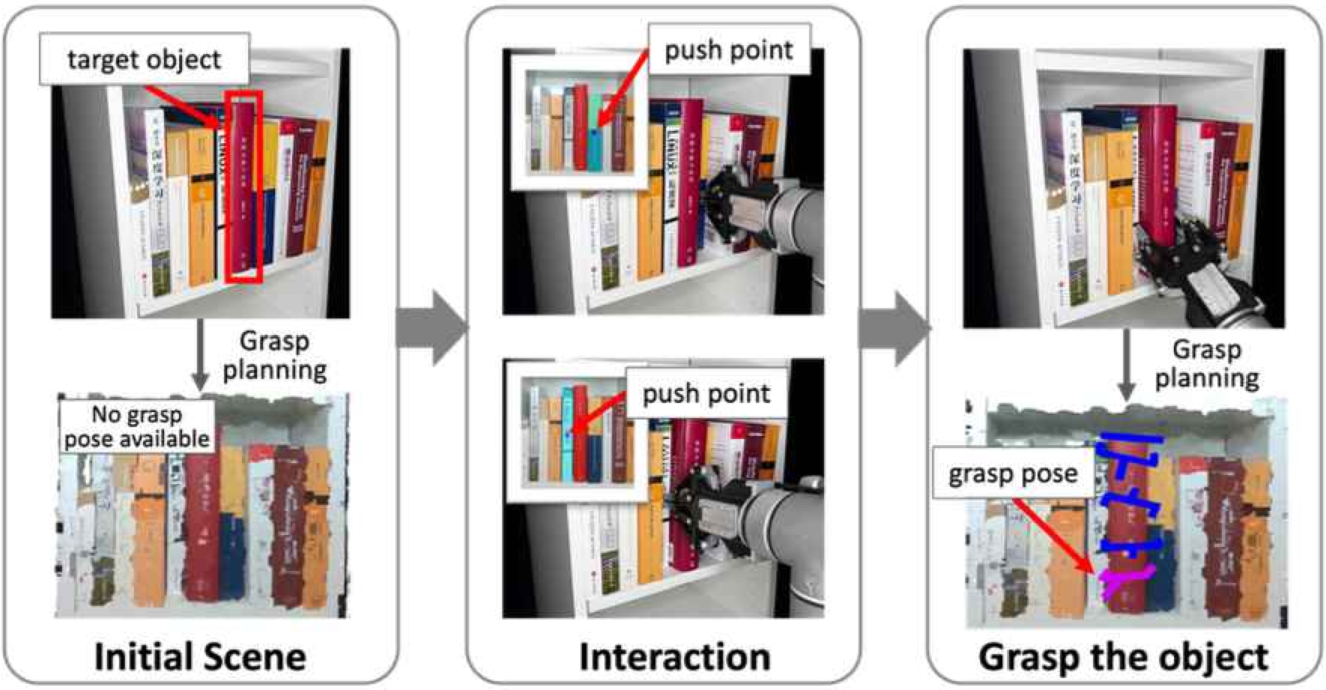


Fig. 1: The robot has to to interact with the environment to grasp a target book from the shelf.

2.3 Action:

An interaction loop framework has been proposed.The robot arm will keep interacting with the environment until space is available for robot arm to reach in and grasp.

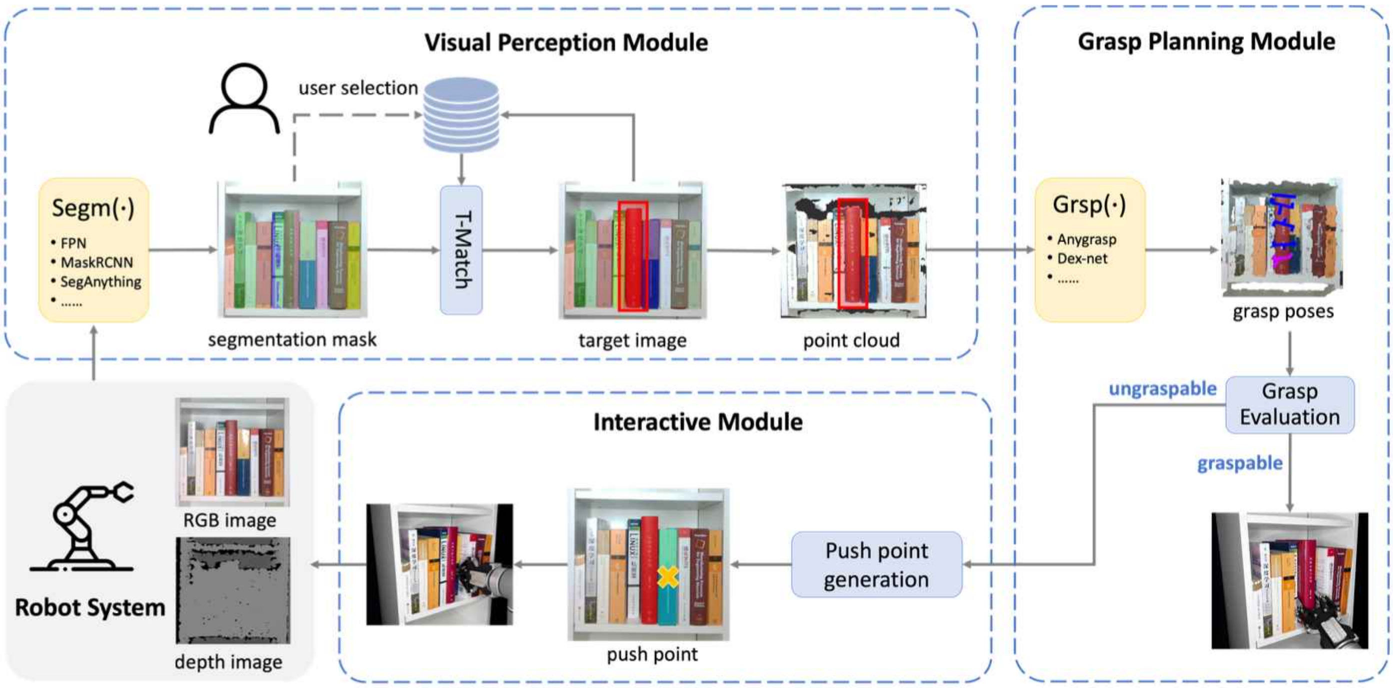
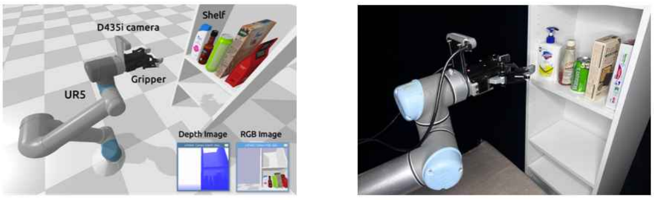


Fig. 2: An overview of the proposed interactive object retrieval adapter.

The simulation environment has been established to evaluate the proposed framework.



(a) simulation environment (b) physical environment

Fig. 4: The overview of the environment.

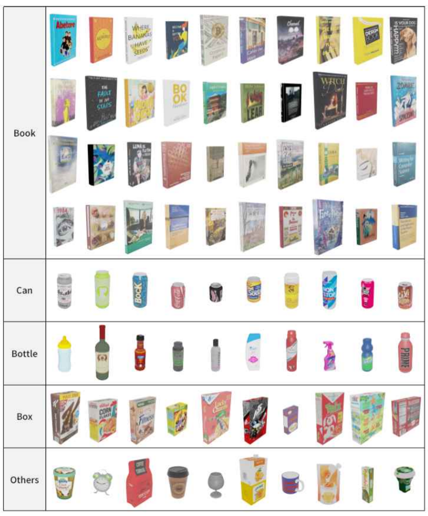


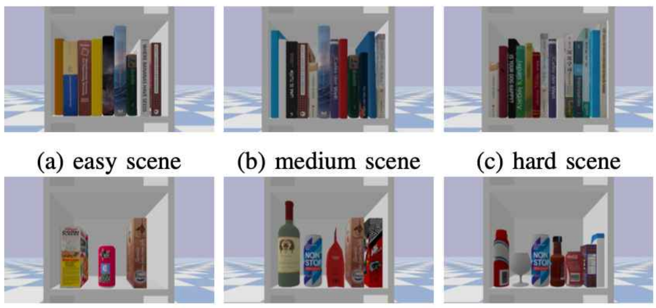
Fig. 5: Selected objects in the dataset.

Utilize MSE to measure the similarity between a mask and target mask.

Algorithm 1 *T-Match*

Input: $Mask, Buffer$
Output: $TargetImage$

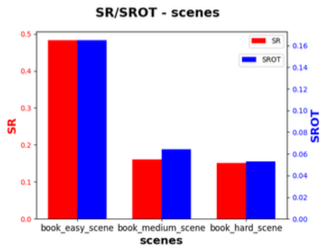
$i = 0$
for *part* **in** $Mask$ **do**
  $score = MSE(part, Buffer)$
  $MScores[i] = score$
  $i = i + 1$
end for
$Index = argmin(MScores)$
$TargetImage = Mask[Index]$
return $TargetImage$



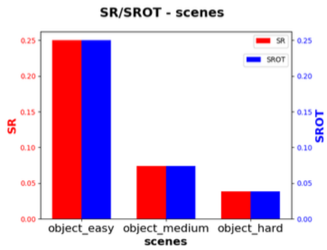
(a) easy scene (b) medium scene (c) hard scene
(d) easy scene (e) medium scene (f) hard scene

Fig. 6: Initial scenes with different difficulty levels.

2.4 Result:

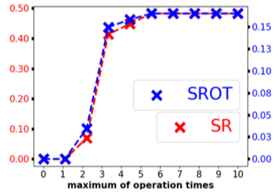


(a) results in Scene 1

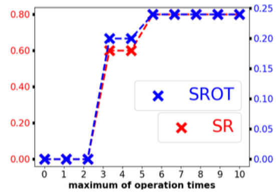


(b) results in Scene 2

In our proposed easy scenario, the robot perform best, while in hard scenario it perform worst



(a) simulation



(b) physical

Fig. 8: The analysis of how **OT** will influence the **SR** and **SROT**.

if we limit the interaction times(OT), the success rate will be reduced

⚽ An illustration of how our framework works.

Fig. 10: The robot interacts with environment to grasp the target object.

TABLE II: Comparison results in simulation environment

methods	SR	OT	SROT
AGRSP	0	0	0
Random	0	2.35	0
Ours(w/o Depth)	0.39	6.09	0.12
Ours(w/o GEval)	0.17	2.2	0.06
Ours	0.48	3.69	0.16

TABLE III: Comparison results in physical environment

methods	SR	OT	SROT
AGRSP	0	0	0
Random	0.6	5.2	0.1
Ours(w/o Depth)	0	10	0
Ours(w/o GEval)	0.4	2.6	0.13
Ours	0.8	4.8	0.24

TABLE I: Results across different scenes in the simulation

	Scene 1			Scene 2		
	easy	medium	hard	easy	medium	hard
SR	0.48	0.16	0.15	0.25	0.07	0.04
OT	3.69	4.23	4.36	1.33	1.74	3.12
SROT	0.16	0.06	0.05	0.25	0.07	0.04

3. Sentiment Classification with CNN and LSTM

> Language Understanding

3.1 Situation:

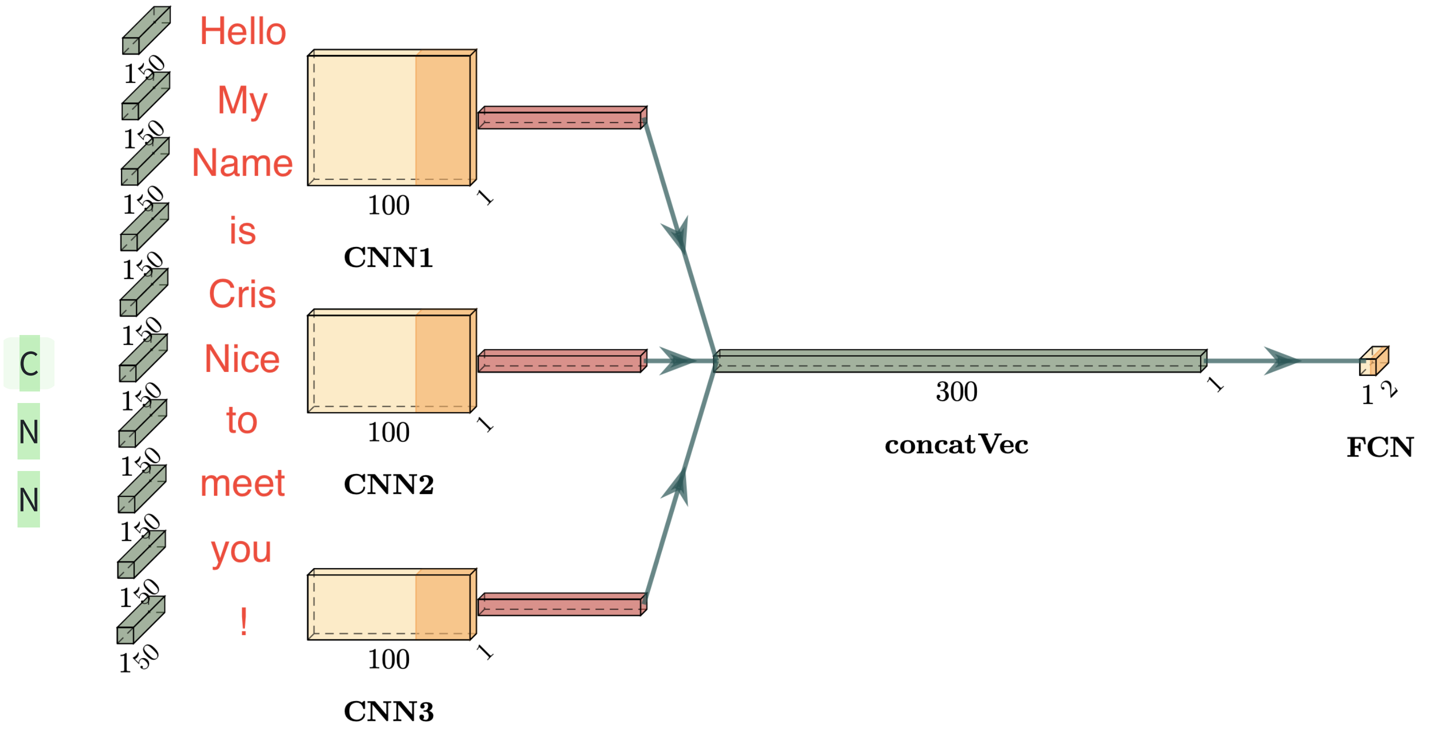
Utilizing deep learning for sentiment classification enables computers to automatically interpret the emotional tone of text, providing valuable insights for businesses to understand customer feedback, personalizing user experiences, and aiding in large-scale data analysis. This automated understanding is crucial for improving products and services, informed decision-making, and conducting social research, making it an essential tool in the realm of natural language processing.

3.2 Target

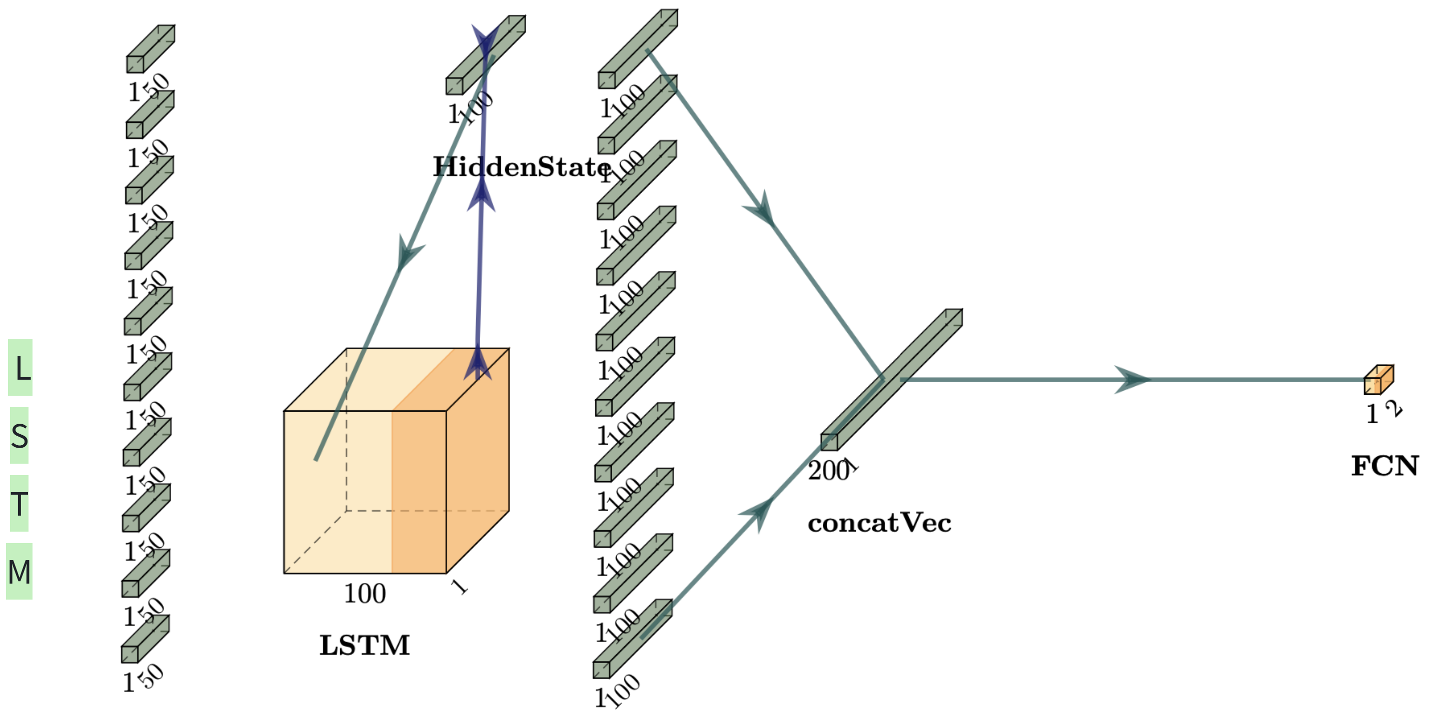
Through designing CNN and LSTM models and training on large labeled datasets, I make a deep learning model capable of predicting the sentiment of a sentence.

	Data	Label
Train	I absolutely loved the friendly service and the breathtaking view at the hotel – it was an unforgettable experience!	😉
	I’m really disappointed with the slow performance and constant crashes of the new update – it’s completely frustrating.	🤬
	......	......
Predict	The movie was a masterpiece – the plot twists were ingenious!	❓
	......	❓

3.3 Action:



Replace the words with embedding vector(dim=50),all of the embeddings form an 2d array with shape [words_num * 50]; then process the sentence with three CNN(bright orange) with kernel size is 3*50,4*50,5*50 respectfully,after that, use relu(dark orange) to activate non-linear atttributes and max pooling(red) to compress features.Concatenate(green) three vectors,feed it into a Linear Layer to get 2 outputs(Positive and Negative score).



Replace the words with embedding vector(dim=50), then process the embedding sequence one by one, with LSTM(orange) model updates a hidden state and outpus a vecotr after processing each embedding,after that, concatenate(green) the first and the last vectors,feed it into a Linear Layer to get 2 outputs(Positive and Negative score).

3.4 Result:

Evaluate CNN and LSTM

Model	CNN	LSTM
ACC	83.47%	84.01%
F-Score	0.84	0.84

Explore How BatchSize will Influence Performance

BatchSize(CNN)	1	25	50	100
ACC	81.57%	80.22%	82.93%	82.93%
F-Score	0.83	0.81	0.83	0.83

---

4. IME(Input Method Editor) with NaiveBayes

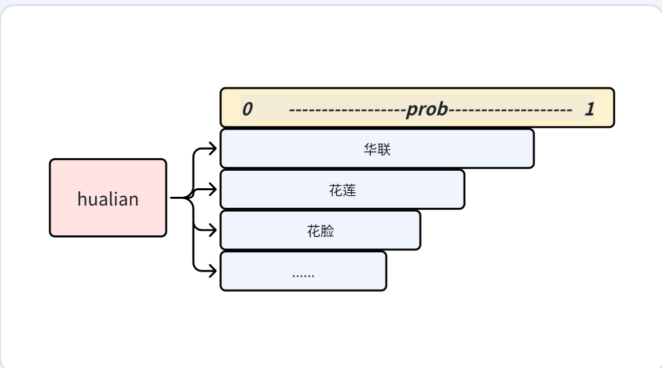
> Language Understanding

4.1 Situation:

The practical significance of Pinyin Input Method Editor (IME) projects is profound, as they greatly enhance work efficiency by enabling users to input Chinese characters quickly and accurately. On the other hand, these projects are rich in core content such as Bayesian posterior probability models, shortest path algorithms, and data preprocessing, which are extremely useful for practicing deep learning. The application of these algorithms and models means that IME projects serve as excellent examples for deepening the understanding and skills of deep learning concepts and techniques. They assist developers and researchers in applying and advancing deep learning technology in real-world scenarios.

4.2 Target

When given a sequence of Pinyin (letters representing Chinese sounds), there are often multiple corresponding Chinese characters or words that it could represent. It is crucial to accurately determine the specific character or word the user intends to write. This task of disambiguation is essential for creating an effective and user-friendly input method.



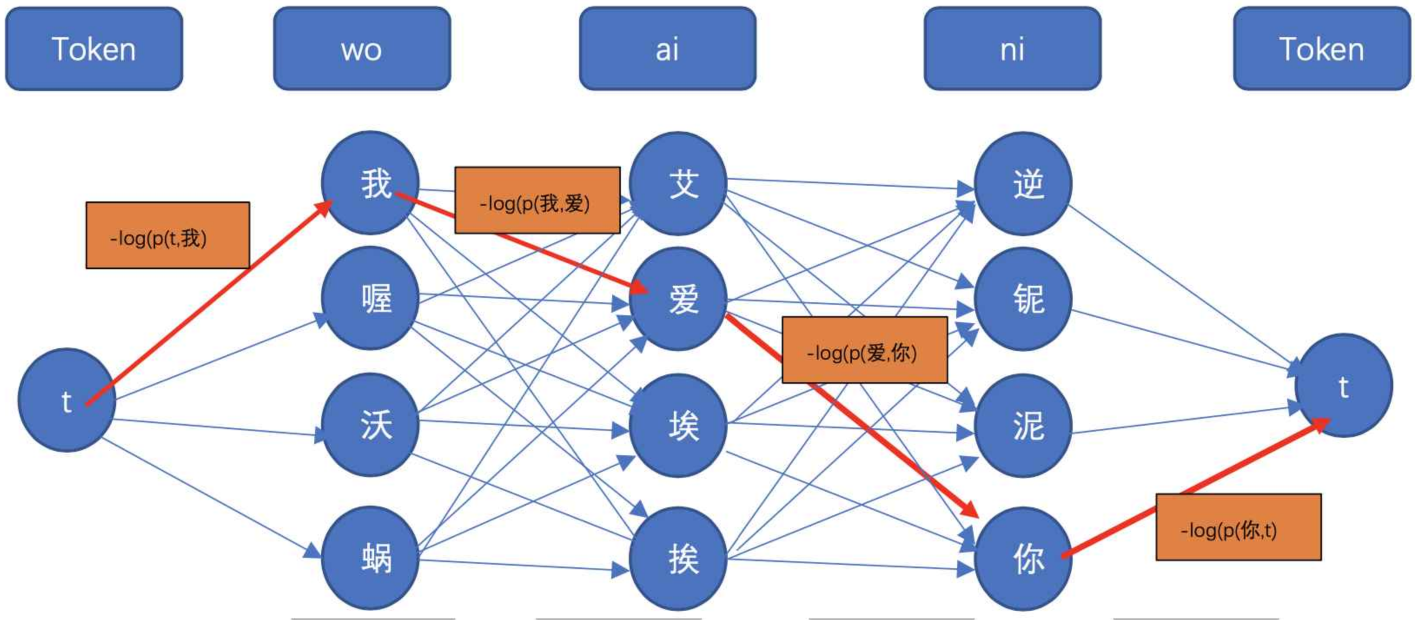
4.3 Action:

In this project, sentences are treated as random variables, with the probability of such a variable being calculated through the chain rule of conditional probabilities that depend on each word and its preceding words. The chain rule is further simplified in this context, based on the assumption that the probability of each Chinese character’ s occurrence is solely dependent on the character that comes before it (a bigram model). This leads to a simplified version of the chain rule. After the Pinyin of a sentence is given, the task is to simply identify the random variable with the highest probability among all possible variables. This variable will be the most likely sentence that corresponds to the given Pinyin.

$$P(w_1, w_2, w_3, \ldots) = P(w_1) \cdot P(w_2|w_1) \cdot P(w_3|w_2) \cdot \ldots \cdot P(w_i|w_{i-1})$$

For efficiency, the multiplication of conditional probabilities in the chain rule is converted to the addition of log conditional probabilities, which makes it suitable for use with the Viterbi algorithm. The problem is thus transformed from finding the combination of Chinese characters with the highest probability to a greedy shortest path search from the initial Pinyin to the ending Pinyin. With the assistance of the Viterbi algorithm, this allows for a more efficient search for the optimal solution.

$$\log P(w_1, w_2, \ldots) = \log P(w_1) + \log P(w_2|w_1) + \log P(w_3|w_2) + \ldots + \log P(w_i|w_{i-1}) + \ldots$$



4.4 Result:

Word Accuracy:0.66	Sentence Accuracy:0.09

5. MNIST Classification with K-means

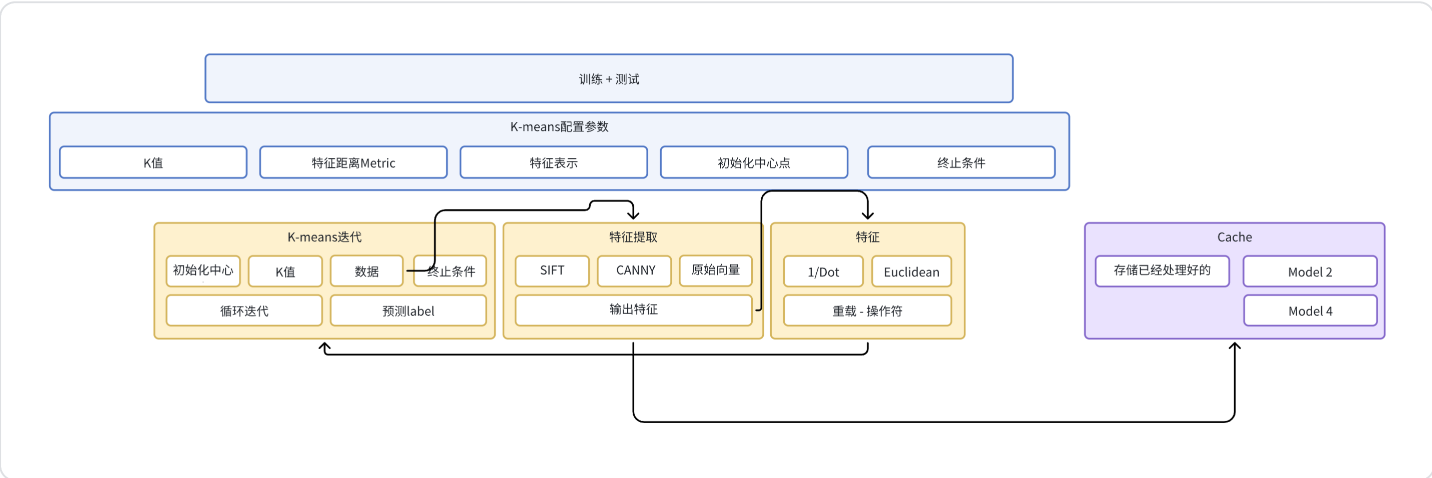
> 2D Image Understanding

5.1 Situation:

The ability to recognize images, particularly handwritten digital images, is a manifestation of intelligence and is one of the essential tasks that artificial intelligence must learn. Unsupervised methods, such as K-means, can save on the cost of labeling while learning from the inherent similarities in the data, thereby clustering the data.

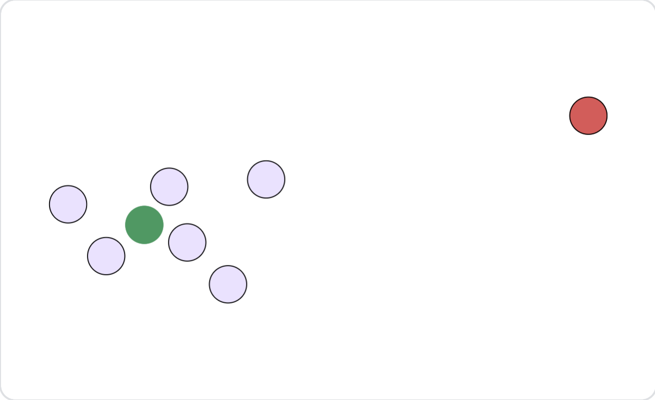
5.2 Target

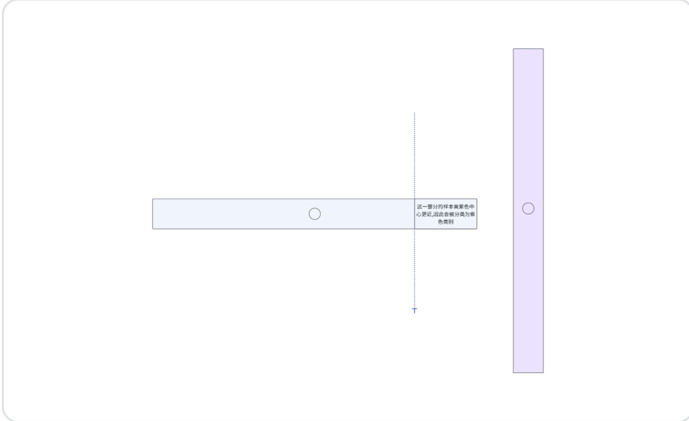
Use K-means Algorithm to cluster digital images, learn their labels, predict the label of new images.



5.3 Action:

1. The MNIST dataset contains 10 classes, so for clustering, different values of K should be selected around K=10, based on a grid search to explore the hyperparameter K’ s value.
2. During the iteration loop, some centroids may become isolated, meaning that no sample belongs to that class. For example, in the figure(left below), according to the nearest neighbor principle, the purple data points would all be classified under the green centroid’ s class, leaving the red centroid as an outlier. So, how should its position be updated next time? Solution: Randomization reduces randomness. Instead of randomly selecting centroids from the entire sample space, centroids are randomly chosen from among the samples, ensuring that each class contains at least one sample.





3. If the distribution of two classes is elongated and perpendicular to each other, even if their respective class representatives are at the center, there is a high probability of misprediction. Solution: During data preprocessing, normalize the distribution across dimensions.

5.4 Result:

Different Features' Performance

feature	OriginalVector	SIFT	CANNY
ACC	0.86	0.35	**0.9**

Which Metric Better?

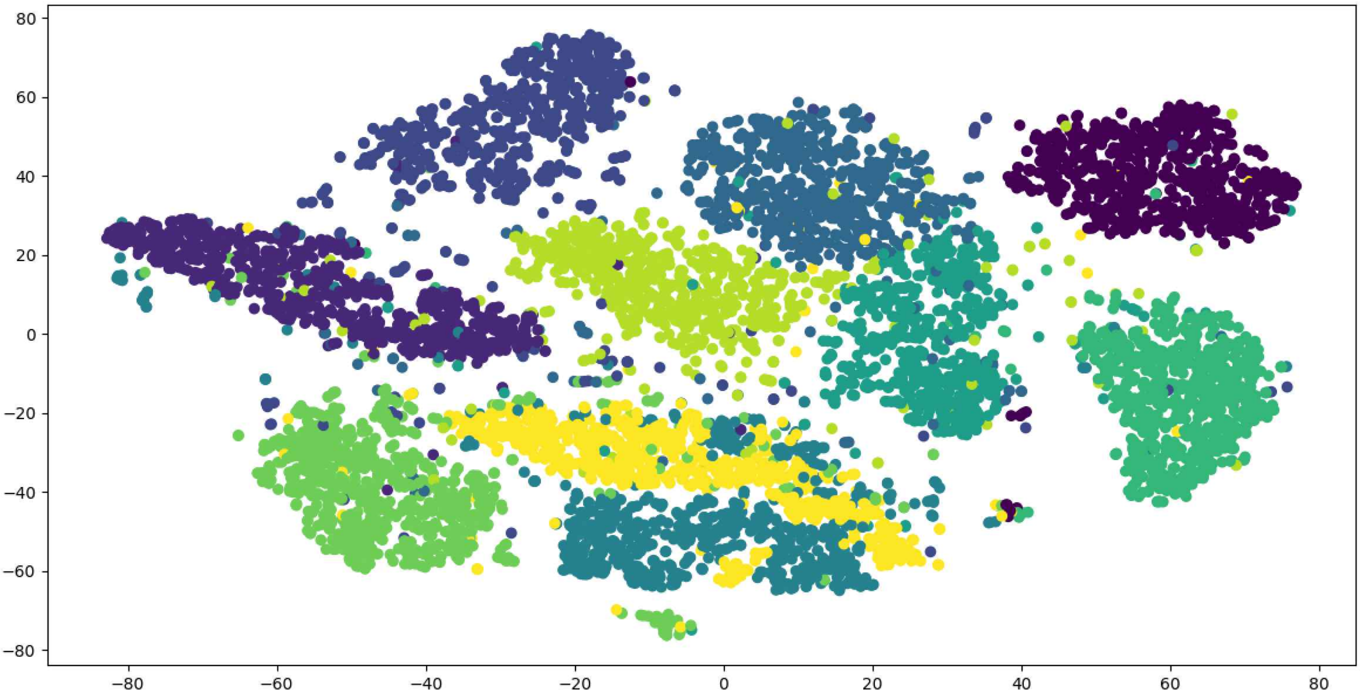
distance_metric	1/DotProduct	Euclidean
ACC	0.51	**0.53**

Normalization Matters?

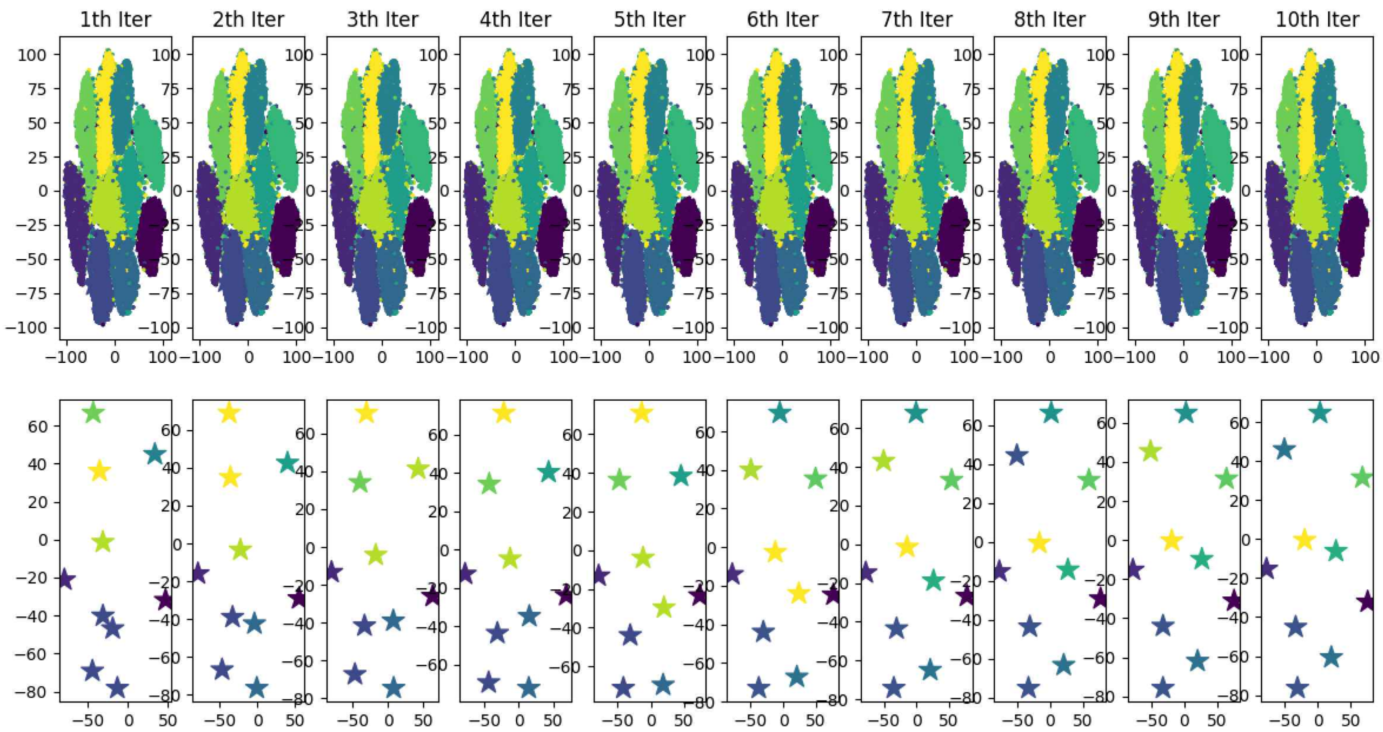
Norm	Norm	w/o
ACC	0.76	**0.85**

Performance-K

K	10	11	12	13	14	15	16	17	20	50
ACC	0.2	0.6	0.6	14	0.6	0.6	0.55	0.6	0.8	**0.9**



Two-dimensional visualization of training data using t-SNE tends to cluster data with the same labels in proximity.



Visualization of the Evolution of K Centroids’ Positions Throughout the Training Process

---

6. SPAM Detector with NaiveBayes

> Language Understanding

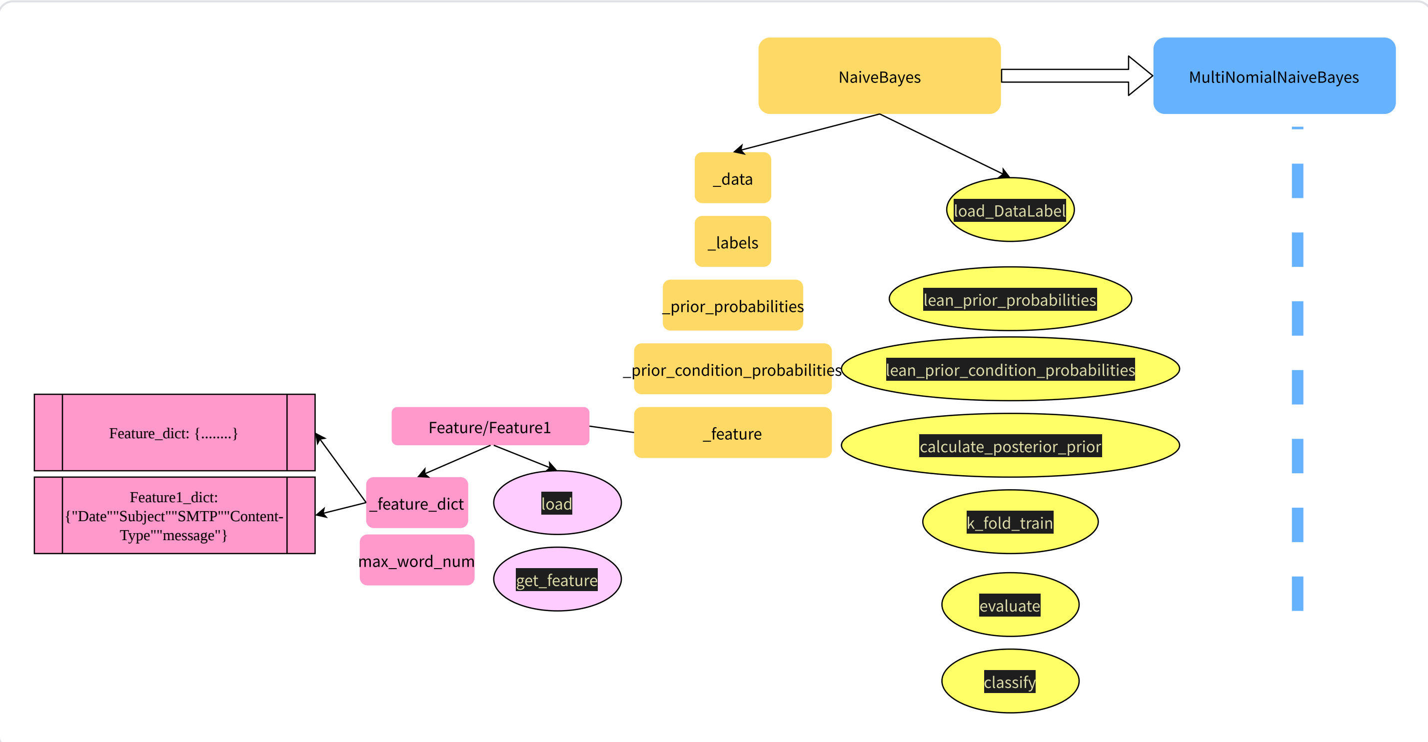
6.1 Situation:

The detection of spam emails can assist individuals in automatically filtering out irrelevant information, thereby enhancing their work efficiency. Additionally, implementing spam detection using the Naive Bayes method holds significant value as it leverages the principles of probability to effectively distinguish between spam and non-spam emails. This approach is not only computationally efficient but also achieves a high level of accuracy with relatively simple mathematical models. By training on a dataset of labeled emails, the Naive Bayes classifier learns the likelihood of certain features associated with spam, which can then be applied to new emails to predict their classification. As a result, it serves as a powerful tool for maintaining the integrity of our inboxes and safeguarding our digital communication.

6.2 Target

Complement NaiveBayes and MultiNomial Bayes models. Train them with spam and ham data.

6.3 Action:



If one were to strictly adhere to the formula for posterior probability, the product of all conditional probabilities would result in an extremely small number (with more than 100 zeros after the decimal point), rendering it impractical to compare magnitudes of probabilities for effective classification.By employing an equivalent transformation, probabilities are converted into logarithmic values, which means that multiplication of probabilities is correspondingly transformed into the addition of log values.

NaiveBayes Learn

$$P(w_1 = \text{num1}, w_2 = \text{num2}, w_3 = \text{num3}, \ldots)$$

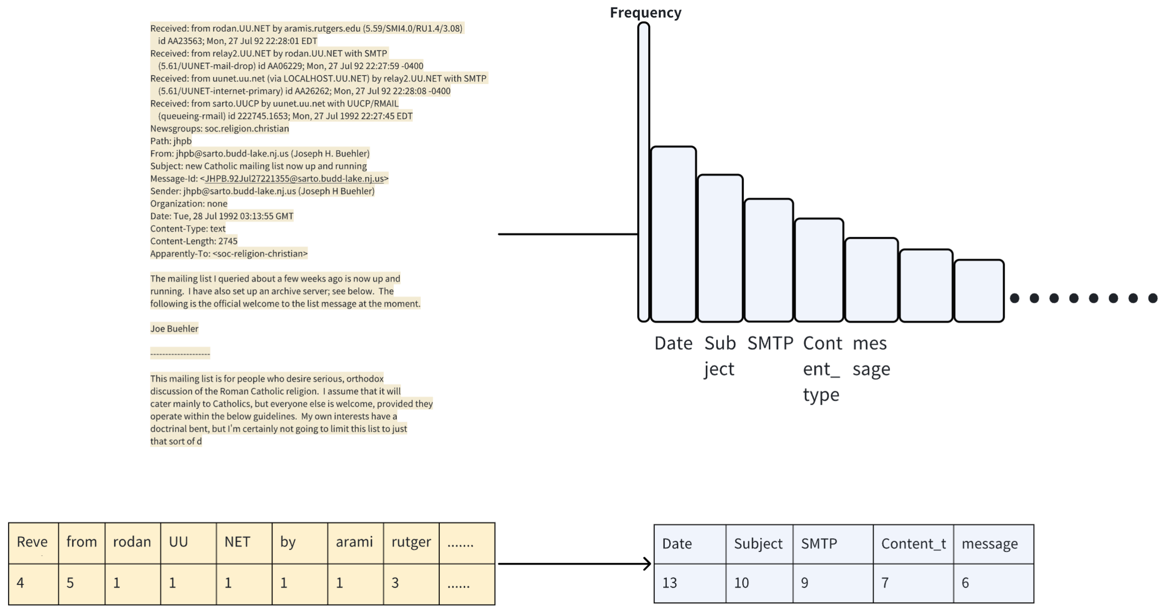
Based on Assumption--None

MultiNomialBayes Learn

$$P(w_1 = \text{num1}, w_2 = \text{num2}, w_3 = \text{num3}, \ldots)$$

Based on Assumption--MultiNomial Distribution:

$$P(w_1 = \text{num1}, w_2 = \text{num2}, w_3 = \text{num3}, \ldots) = P(w_1)^{\text{num1}} \cdot P(w_2)^{\text{num2}} \cdot P(w_3)^{\text{num3}} \ldots$$



Full Text Feature

Five Words Feature(Feature1)

6.4 Result:

alpha	0.01	0.1	1	5	10
ACC	***0.985***	0.553	0.551	0.551	0.551
F1	***0.983***	0.711	0.710	0.710	0

feature	Feature	Feature1
ACC	***0.982***	0.807
F1	***0.980***	0.844

train_size	20%	40%	60%	80%	100%
ACC	0.886	0.815	0.907	0.975	***0.982***
F1	0.780	0.727	0.880	0.977	***0.980***

Model	NB	Multi Nomial
ACC	0.982	***1.0***
F1	0.980	***1.0***

⚽ In the context of Naive Bayes classification, the problem described is known as "zero-frequency" and can severely affect the performance of the classifier. When no instances of a particular class and feature value combination have been observed in the training data, the estimated probability for that combination will be zero. This can lead to a situation where the posterior probability of a class given an observation is zero, essentially ruling out that class regardless of the evidence provided by other features.

To mitigate this issue, a technique called "Laplace smoothing" (or "additive smoothing") is applied. This involves adding a small positive value (alpha) to the count for each class-feature combination when calculating probabilities. This adjustment ensures that no probability is ever exactly zero and allows for a more robust comparison between classes even when dealing with previously unobserved feature values.

Here's how Laplace smoothing is incorporated into the probability calculation:

For a given feature (x_i) and class (c), instead of calculating the conditional probability as

$$(P(x_i = k|y = c) = \frac{\text{Number of times } x_i = k \text{ in class } c}{\text{Total number of samples in class } c})$$

, with Laplace smoothing it becomes:

$$[P(x_i = k|y = c) = \frac{\text{Number of times } x_i = k \text{ in class } c + alpha}{\text{Total number of samples in class } c + K * alpha}]$$

where (K) is the number of distinct values that the feature (x_i) can take on.

When using log probabilities, the log of zero is not defined, which is another reason why smoothing is necessary. After smoothing, since all probabilities are non-zero, their logarithms will be well-defined real numbers, allowing for the log probabilities to be summed without issue.