

	KNN	SVM
Training Speed	~ 0.5 -1.5 min	~ 12 - 19 min
preprocessing	Use scaler(normalizer), pca	scaler(standard scaler)
Feature extraction	Hand-crafted (HOG, LBP, Color Histograms) And then combine them into one histogram And then use PCA to shrink high variance	Deep Learning Features (VGG16 Embeddings) Model learned how to recognize complex patterns and we took the last layer which in 512 vector
Threshold used	0.6	0.44
Parameters	n_neighbors =5 metric = 'euclidian'	C= 100, kernel='rbf', class_weight ='balanced'
memory	Store the entire training set	Only stores support vectors
Accuracy	<ul style="list-style-type: none"> ● On Augment on whole data : 0.86 ● When augment on only training data : 0.59 ● When improving feature extraction (changing hyper parmeter and model parameter) : 0.63 ● After Using CNN on feature extraction : 0.86 	<ul style="list-style-type: none"> ● On Augment on whole data : 0.84 ● When augment on only training data : 0.55 ● When improving feature extraction (changing hyper parmeter and model parameter) : 0.67 ● After Using CNN on feature extraction : 0.85

Final Improved KNN Accuracy :

```
KNN Results
Standard Accuracy (No Rejection): 0.8587
Accuracy on Accepted Samples: 0.8837
Rejection Rate: 3.73%

Classification Report (Standard):
```

	precision	recall	f1-score	support
cardboard	0.98	0.86	0.91	50
glass	0.81	0.84	0.83	77
metal	0.82	0.89	0.85	63
paper	0.90	0.91	0.91	90
plastic	0.87	0.82	0.85	73
trash	0.70	0.73	0.71	22
accuracy			0.86	375
macro avg	0.85	0.84	0.84	375
weighted avg	0.86	0.86	0.86	375

Final improved SVM Accuracy:

Training SVM

SVM Test Accuracy: 0.8507

SVM Rejection Rate: 2.13%

D:\Anaconda\Lib\site-packages\sklearn\metrics_classification.py:1569
bels with no true samples. Use `zero_division` parameter to control t
_warn_prf(average, modifier, f"{metric.capitalize()} is", len(resu
D:\Anaconda\Lib\site-packages\sklearn\metrics_classification.py:1569
bels with no true samples. Use `zero_division` parameter to control t
_warn_prf(average, modifier, f"{metric.capitalize()} is", len(resu
D:\Anaconda\Lib\site-packages\sklearn\metrics_classification.py:1569
bels with no true samples. Use `zero_division` parameter to control t
_warn_prf(average, modifier, f"{metric.capitalize()} is", len(resu

Classification Report:

	precision	recall	f1-score	support
0	0.95	0.80	0.87	50
1	0.87	0.87	0.87	77
2	0.85	0.92	0.89	63
3	0.87	0.94	0.90	90
4	0.87	0.81	0.84	73
5	0.71	0.45	0.56	22
6	0.00	0.00	0.00	0
accuracy			0.85	375
macro avg	0.73	0.69	0.70	375
weighted avg	0.87	0.85	0.86	375