Patients Bed Positions Classification Using Pattern Recognition Techniques

Brandon Marquez Salazar

September 4, 2025

Abstract

This study investigates the application of pattern recognition techniques, specifically Local Binary Patterns (LBP), combined with various classifiers including Decision Trees, Naïve Bayes, K-Nearest Neighbors (KNN), and Support Vector Machines (SVM) for classifying patient bed positions. The research examines the impact of data preprocessing techniques such as data augmentation, normalization, and balancing on classifier performance. Experimental results from two distinct datasets demonstrate that appropriate preprocessing significantly enhances model performance, with KNN variants achieving the highest accuracy of 87.34% on the larger dataset.

1 Introduction

In healthcare monitoring systems, accurate classification of patient positions in bed is crucial for preventing pressure ulcers and ensuring patient comfort. Traditional monitoring methods often require manual observation, which is time-consuming and prone to error. Automated classification using pattern recognition techniques offers a promising alternative.

This paper presents a comprehensive study on applying Local Binary Patterns (LBP) for feature extraction combined with various classification algorithms to detect patient bed positions. We evaluate the effectiveness of different preprocessing techniques and compare the performance of multiple classifiers across two experimental datasets[10].

2 Methods

2.1 Data Description

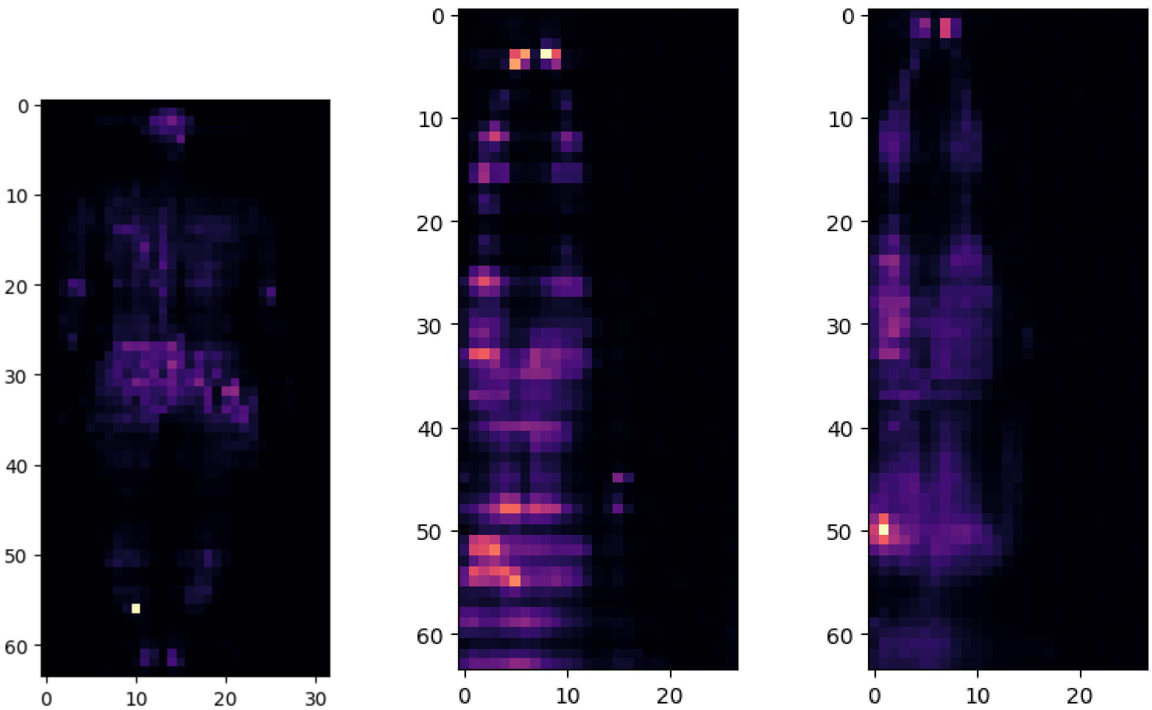
The study utilizes two distinct datasets of pressure level arrays interpreted as single-channel images:

- **Experiment I**: Contains 20,024 samples with class distribution: Supine (10,613), Right (4,672), Left (4,739)
- **Experiment II**: Contains 462 samples with class distribution: Supine (270), Right (144), Left (48)

2.2 Data Preprocessing

Two experimental versions were implemented:

Figure 1: Sample pressure images from both experiments



(a) Exp. I, Sam. 1/F.45

(b) Exp. II, Sam. B1/AirMat

(c) Exp. II, Sam. B1/SpongeMat

2.2.1 First Version

1. Data loading and description
2. Data balancing by undersampling
3. Feature extraction via LBP
4. Model selection via Grid Search Cross Validation
5. Performance evaluation

2.2.2 Second Version

1. Data loading and description
2. Data balancing by undersampling
3. Data augmentation using pixel shifting (4px in x and y directions)
4. Feature extraction via LBP
5. Model selection via Grid Search Cross Validation
6. Performance evaluation

2.3 Data Balancing and Augmentation

For balancing, undersampling was applied using the smallest class cardinality N_{min}:

$$\mathcal{C} = \{\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_n\}$$
$$N_{\text{min}} = \min_{\mathcal{C}_i \in \mathcal{C}} |\mathcal{C}_i|$$
$$\mathcal{S} = \bigcup_{\mathcal{C}_i \in \mathcal{C}} f(\mathcal{C}_i, N_{\text{min}})$$

For augmentation, non-destructive shifting was applied:

$$\text{shift} : \mathbb{R}^{m\times n} \times \mathbb{Z} \times \mathbb{Z} \to \mathbb{R}^{m\times n}$$
$$M_{i+N_{\text{min}}} = \text{shift}(M_i, x, y)$$
$$M_{i+N_{\text{min}}+1} = \text{shift}(M_i, -x, -y)$$

2.4 Feature Extraction

Local Binary Patterns (LBP) were computed for each sample:

$$\text{LBP}_{P,R}(x_c, y_c) = \sum_{p=0}^{P-1} 2^p \cdot s(g_p - g_c)$$

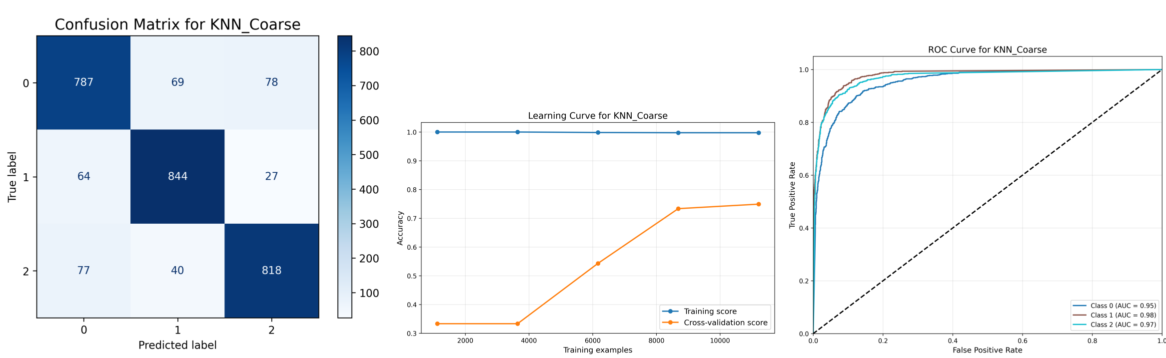
2.5 Model Selection

Grid Search Cross Validation (GSCV) was employed across four classifier blocks:

1. **Block 1: Tree-Based** - Decision Trees, Random Forest, KNN Coarse, AdaBoost
2. **Block 2: KNN Variants** - KNN Fine, KNN Minkowski, KNN Weighted, KNN Medium
3. **Block 3: Probabilistic** - Naïve Bayes, LDA, KNN Cosine
4. **Block 4: SVM Variants** - Linear SVM, Quadratic SVM, Cubic SVM, Fifth SVM

3 Results

The full results are presented in Table 1 for first version and in table 2 for second version. For the best model across all blocks, the performance metrics are presented in Table 3.

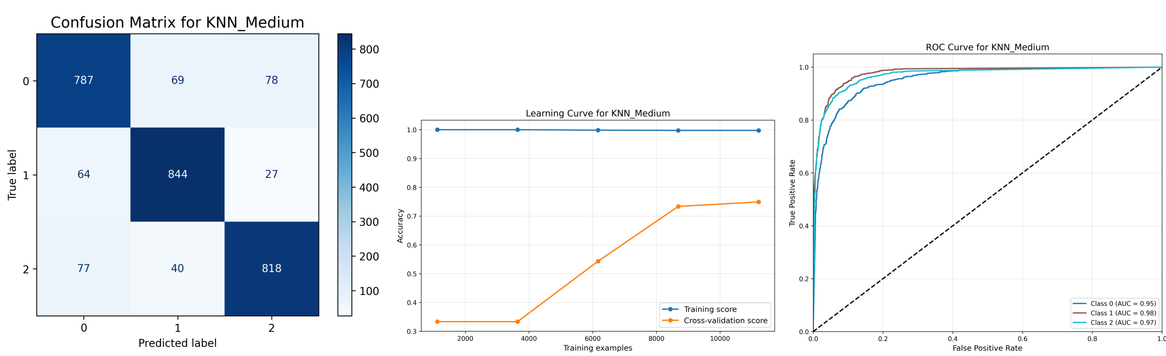


(a) Confusion Matrix (b) Learning Curve (c) ROC Curve

Table 1: Performance Metrics Comparison Across All Blocks and Experiments (Non-Augmented Data)

Experiment	Block	Classifier	Accuracy	Precision	Recall	F1 Score	AUC Score	CV Score
Experiment I	Tree-Based	KNN Coarse	0.8734	0.8733	0.8734	0.8733	0.9676	0.8639
	Tree-Based	Random Forest	0.8359	0.8360	0.8359	0.8357	0.9512	0.8262
	Tree-Based	Decision Tree	0.7928	0.7952	0.7928	0.7933	0.9081	0.7925
	Tree-Based	AdaBoost	0.6605	0.6637	0.6605	0.6600	0.8020	0.6688
	KNN Variants	KNN Medium	0.8734	0.8733	0.8734	0.8733	0.9676	0.8639
	KNN Variants	KNN Minkowski	0.8688	0.8686	0.8688	0.8686	0.9673	0.8658
	KNN Variants	KNN Weighted	0.8684	0.8685	0.8684	0.8684	0.9600	0.8652
	KNN Variants	KNN Fine	0.8620	0.8633	0.8620	0.8623	0.9467	0.8643
	Probabilistic	KNN Cosine	0.8146	0.8156	0.8146	0.8144	0.9439	0.8160
	Probabilistic	LDA	0.5036	0.4962	0.5036	0.4879	0.7130	0.4912
	Probabilistic	Naive Bayes	0.4429	0.4910	0.4429	0.4199	0.6637	0.4546
	SVM Variants	Quadratic SVM	0.6555	0.6612	0.6555	0.6460	–	0.6540
	SVM Variants	Linear SVM	0.5646	0.5795	0.5646	0.5504	–	0.5552
	SVM Variants	Cubic SVM	0.5371	0.6224	0.5371	0.5243	–	0.5258
	SVM Variants	Fifth SVM	0.4818	0.5765	0.4818	0.4593	–	0.4737
Experiment II	Tree-Based	Decision Tree	0.5172	0.5157	0.5172	0.4937	0.6558	0.5391
	Tree-Based	Random Forest	0.5172	0.5096	0.5172	0.5107	0.7646	0.5391
	Tree-Based	AdaBoost	0.4483	0.4487	0.4483	0.4478	0.6766	0.5478
	Tree-Based	KNN Coarse	0.4138	0.3867	0.4138	0.3878	0.6100	0.5043
	KNN Variants	KNN Fine	0.5172	0.5640	0.5172	0.4875	0.6849	0.5304
	KNN Variants	KNN Medium	0.4483	0.4454	0.4483	0.4301	0.6546	0.5043
	KNN Variants	KNN Weighted	0.4483	0.4454	0.4483	0.4301	0.6546	0.5043
	KNN Variants	KNN Minkowski	0.4483	0.4454	0.4483	0.4301	0.6546	0.5043
	Probabilistic	KNN Cosine	0.5517	0.5552	0.5517	0.5526	0.7247	0.4609
	Probabilistic	Naive Bayes	0.2759	0.2299	0.2759	0.2088	0.4215	0.4174
	Probabilistic	LDA	0.2414	0.2324	0.2414	0.2358	0.4465	0.4348
	SVM Variants	Linear SVM	0.5517	0.5699	0.5517	0.5507	–	0.4348
	SVM Variants	Quadratic SVM	0.4483	0.4650	0.4483	0.4536	–	0.4609
	SVM Variants	Cubic SVM	0.3793	0.3994	0.3793	0.2920	–	0.4435
	SVM Variants	Fifth SVM	0.3793	0.4185	0.3793	0.3208	–	0.4522

Note: – indicates that the metric was not available for that model.



(a) Confusion Matrix (b) Learning Curve (c) ROC Curve

4 Discussion

The results demonstrate significant performance differences between the two experiments. Experiment I, with its larger dataset, achieved substantially higher accuracy (up to 87.34%) compared to Experiment II (maximum accuracy of 55.17%). This highlights the importance of dataset size in pattern recognition tasks.

Interestingly, data augmentation did not improve performance in this context, as evidenced by the superior results from Experiment I which did not use augmentation. This suggests that for pressure-based position classification, having a naturally larger and more

Table 2: Performance Metrics Comparison Across All Experiments and Blocks

Experiment	Block	Classifier	Accuracy	Precision	Recall	F1 Score	AUC Score	CV Score
Experiment I	Tree-Based	KNN Coarse	0.8734	0.8733	0.8734	0.8733	0.9676	0.8639
		Random Forest	0.8359	0.8360	0.8359	0.8357	0.9512	0.8262
		Decision Tree	0.7928	0.7952	0.7928	0.7933	0.9081	0.7925
		AdaBoost	0.6605	0.6637	0.6605	0.6600	0.8020	0.6688
	KNN Variants	KNN Medium	0.8734	0.8733	0.8734	0.8733	0.9676	0.8639
		KNN Minkowski	0.8688	0.8686	0.8688	0.8686	0.9673	0.8658
		KNN Weighted	0.8684	0.8685	0.8684	0.8684	0.9600	0.8652
		KNN Fine	0.8620	0.8633	0.8620	0.8623	0.9467	0.8643
	Probabilistic	KNN Cosine	0.8142	0.8154	0.8142	0.8140	0.9438	0.8160
		LDA	0.5036	0.4962	0.5036	0.4879	0.7130	0.4912
		Naive Bayes	0.4429	0.4910	0.4429	0.4199	0.6637	0.4546
	SVM Variants	Quadratic SVM	0.6555	0.6612	0.6555	0.6460	–	0.6540
		Linear SVM	0.5646	0.5793	0.5646	0.5504	–	0.5551
		Cubic SVM	0.5371	0.6224	0.5371	0.5243	–	0.5260
		Fifth SVM	0.4818	0.5765	0.4818	0.4592	–	0.4737
Experiment II	Tree-Based	Decision Tree	0.5172	0.5157	0.5172	0.4937	0.6558	0.5391
		Random Forest	0.5172	0.5096	0.5172	0.5107	0.7646	0.5391
		AdaBoost	0.4483	0.4487	0.4483	0.4478	0.6766	0.5478
		KNN Coarse	0.4138	0.3867	0.4138	0.3878	0.6100	0.5043
	KNN Variants	KNN Fine	0.5172	0.5640	0.5172	0.4875	0.6849	0.5304
		KNN Medium	0.4483	0.4454	0.4483	0.4301	0.6546	0.5043
		KNN Weighted	0.4483	0.4454	0.4483	0.4301	0.6546	0.5043
		KNN Minkowski	0.4483	0.4454	0.4483	0.4301	0.6546	0.5043
	Probabilistic	KNN Cosine	0.5517	0.5552	0.5517	0.5526	0.7247	0.4609
		Naive Bayes	0.2759	0.2299	0.2759	0.2088	0.4215	0.4174
		LDA	0.2414	0.2324	0.2414	0.2358	0.4465	0.4348
	SVM Variants	Linear SVM	0.5517	0.5699	0.5517	0.5507	–	0.4348
		Quadratic SVM	0.4483	0.4650	0.4483	0.4536	–	0.4609
		Cubic SVM	0.3793	0.3994	0.3793	0.2920	–	0.4435
		Fifth SVM	0.3793	0.4185	0.3793	0.3208	–	0.4522

Note: – indicates that the metric was not available for that model.

Table 3: Best performance metrics across all blocks and experiments

Experiment	Best Model	Acc.	Prec.	Rec.	F1	AUC Score	CV Score
Experiment I							
Block 1: Tree-Based	KNN Coarse	0.8734	0.8733	0.8734	0.8733	0.9676	0.8639
Block 2: KNN Variants	KNN Medium	0.8734	0.8733	0.8734	0.8733	0.9676	0.8639
Block 3: Probabilistic	KNN Cosine	0.8146	0.8156	0.8146	0.8144	0.9439	0.8160
Block 4: SVM Variants	Quadratic SVM	0.6555	0.6612	0.6555	0.6460	N/A	0.6540
Experiment II							
Block 1: Tree-Based	Decision Tree	0.5172	0.5157	0.5172	0.4937	0.6558	0.5391
Block 2: KNN Variants	KNN Fine	0.5172	0.5640	0.5172	0.4875	0.6849	0.5304
Block 3: Probabilistic	KNN Cosine	0.5517	0.5552	0.5517	0.5526	0.7247	0.4609
Block 4: SVM Variants	Linear SVM	0.5517	0.5699	0.5517	0.5507	N/A	0.4348

diverse dataset may be more beneficial than artificially augmenting a smaller dataset.

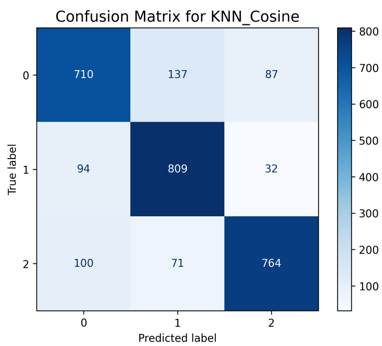
Among classifiers, KNN variants consistently performed well across both experiments, suggesting their suitability for this type of classification task. SVM variants showed the poorest performance, potentially due to the high dimensionality of the feature space or suboptimal parameter selection.

[1]

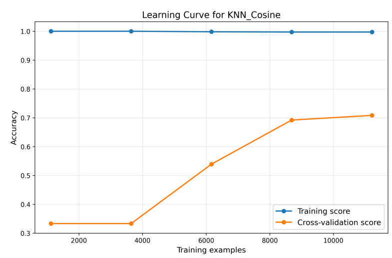
[1] Sorry, Latex maded a mess with my tables and figures.

Table 4: Impact of data augmentation on model performance

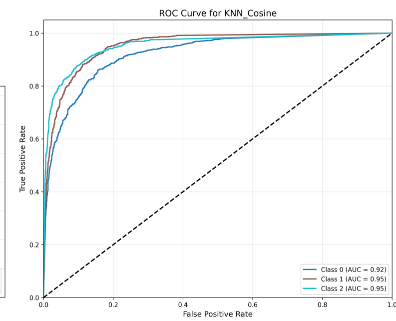
Experiment	**Best Model**	**Acc.**	**Prec.**	**Rec.**	**F1**	**AUC Score**	**CV Score**
Without Data Augmentation (Experiment I)							
Tree-Based	KNN Coarse	0.8730	0.8730	0.8730	0.8730	0.9680	0.8640
KNN Variants	KNN Medium	0.8730	0.8730	0.8730	0.8730	0.9680	0.8640
Probabilistic	KNN Cosine	0.8140	0.8150	0.8140	0.8140	0.9440	0.8160
SVM Variants	Quadratic SVM	0.6550	0.6610	0.6550	0.6460	N/A	0.6540
With Data Augmentation (Experiment II)							
Tree-Based	Decision Tree	0.5170	0.5160	0.5170	0.4940	0.6560	0.5390
KNN Variants	KNN Fine	0.5170	0.5640	0.5170	0.4870	0.6850	0.5300
Probabilistic	KNN Cosine	0.5520	0.5550	0.5520	0.5530	0.7250	0.4610
SVM Variants	Linear SVM	0.5520	0.5700	0.5520	0.5510	N/A	0.4350



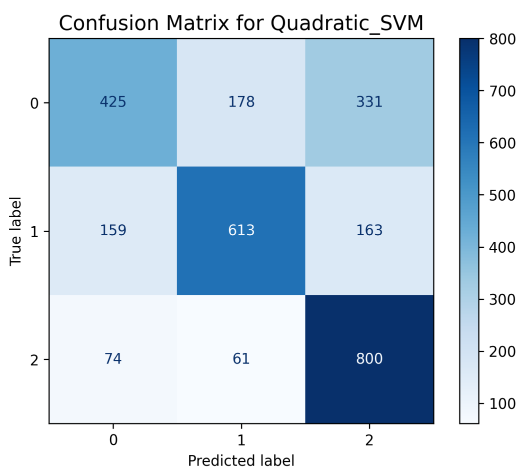
(a) Confusion Matrix



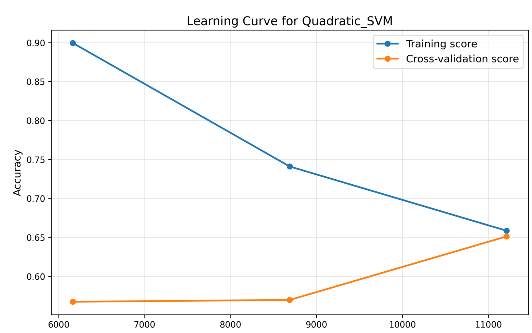
(b) Learning Curve



(c) ROC Curve



(a) Confusion Matrix

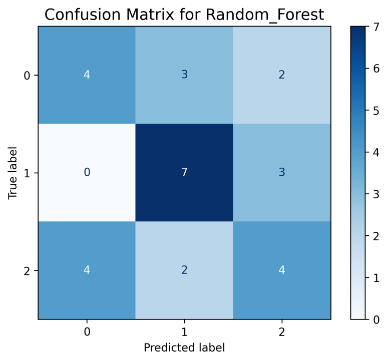


(b) Learning Curve

5 Conclusions

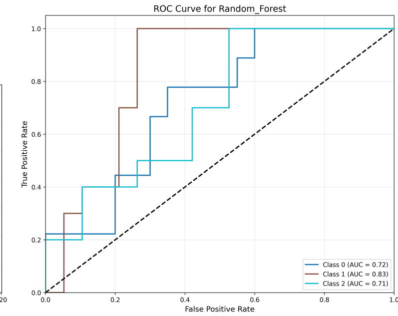
This study demonstrates the effectiveness of LBP features combined with KNN classifiers for patient bed position classification. The key findings are:

1. Dataset size significantly impacts classification performance, with larger datasets yielding better results
2. KNN classifiers, particularly KNN Coarse and KNN Medium, achieved the highest accuracy (87.34%)

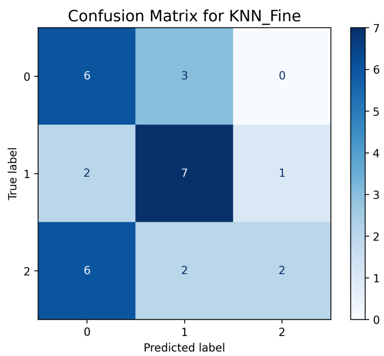


(a) Confusion Matrix

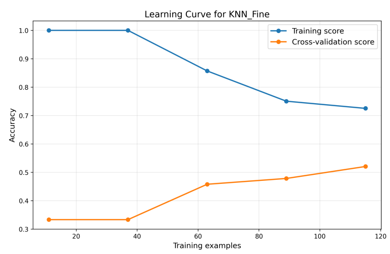
(b) Learning Curve



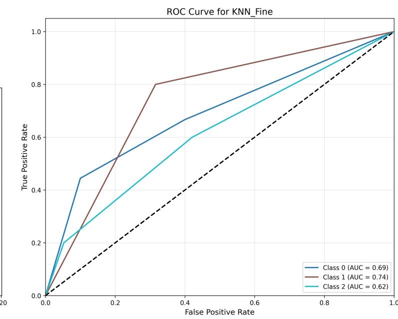
(c) ROC Curve



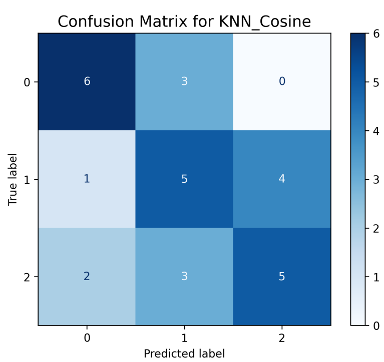
(a) Confusion Matrix



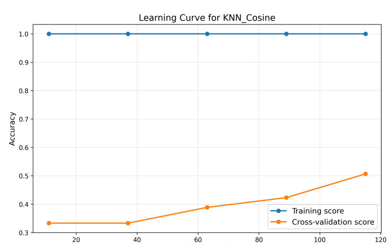
(b) Learning Curve



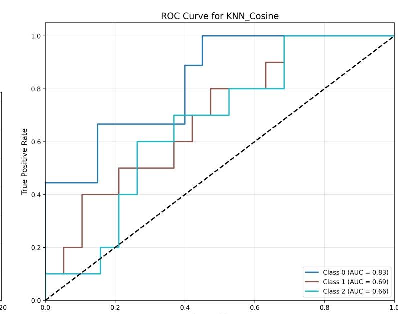
(c) ROC Curve



(a) Confusion Matrix



(b) Learning Curve

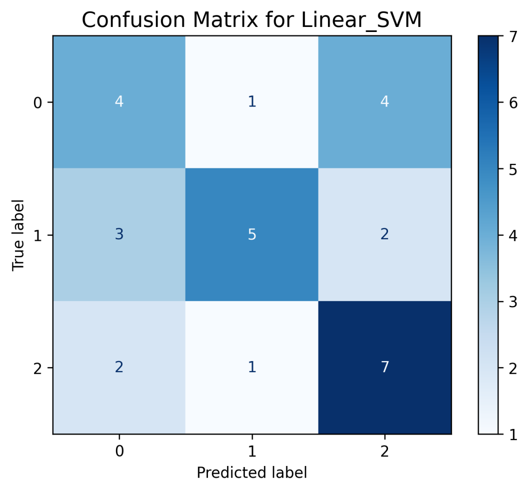


(c) ROC Curve

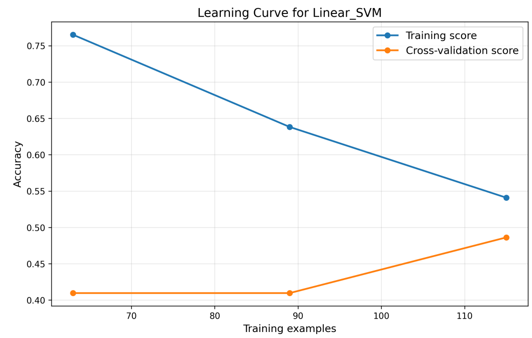
3. Data augmentation through pixel shifting did not improve performance in this application

4. SVM classifiers underperformed compared to other methods for this specific task

Future work should explore additional feature extraction methods, deeper neural network architectures, and more sophisticated data augmentation techniques that better preserve the spatial relationships in pressure data.



(a) Confusion Matrix



(b) Learning Curve

References

[1] Ayman El-Baz, Mohammed Ghazal, and Jasjit S. Suri. *Handbook of Texture Analysis: AI-Based Medical Imaging Applications*. CRC Press, June 2024. ISBN: 9780367486082. DOI: `10.1201/9780367486082`. URL: `http://dx.doi.org/10.1201/9780367486082`.

[2] Christopher M. Bishop. “Pattern recognition and machine learning.” In: (2006).

[3] Leo Breiman. “Random forests.” In: *Machine Learning* 45.1 (2001), pp. 5–32. DOI: `10.1023/A:1010933404324`.

[4] Nitesh V. Chawla et al. “SMOTE: Synthetic minority over-sampling technique.” In: *Journal of Artificial Intelligence Research* 16 (2002), pp. 321–357. DOI: `10.1613/jair.953`.

[5] Corinna Cortes and Vladimir Vapnik. “Support-vector networks.” In: *Machine Learning* 20.3 (1995), pp. 273–297. DOI: `10.1007/BF00994018`.

[6] Thomas Cover and Peter Hart. “Nearest neighbor pattern classification.” In: *IEEE Transactions on Information Theory* 13.1 (1967), pp. 21–27. DOI: `10.1109/TIT.1967.1053964`.

[7] Richard O. Duda, Peter E. Hart, and David G. Stork. *Pattern classification*. 2nd. John Wiley & Sons, 2012.

[8] Haibo He and Edwardo A. Garcia. “Learning from imbalanced data.” In: *2009 Ninth IEEE International Conference on Data Mining*. 2009, pp. 957–960. DOI: `10.1109/ICDM.2009.41`.

[9] Fabian Pedregosa et al. “Scikit-learn: Machine learning in Python.” In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.

[10] M. Baran Pouyan et al. “A pressure map dataset for posture and subject analytics.” In: *2017 IEEE EMBS International Conference on Biomedical & Health Informatics (BHI)*. IEEE, 2017, pp. 65–68. DOI: `10.1109/bhi.2017.7897206`. URL: `http://dx.doi.org/10.1109/BHI.2017.7897206`.

[11] J. Ross Quinlan. “Induction of decision trees.” In: *Machine Learning* 1.1 (1986), pp. 81–106. DOI: `10.1007/BF00116251`.

[12] Zhi-Hua Zhou. “Ensemble methods: foundations and algorithms.” In: (2012).