Deep Learning Vs Machine Learning: A Study on Emotion Classification using both Deep Learning and Machine Learning

Harmouche .O
Electrical, Computer and Biomedical Engineering
Toronto Metropolitan University
Toronto, Ca
oharmouche@torontomu.ca

***Abstract*—This work extends our previous study of speech emotion recognition by presenting a systematic comparison of support vector machines (SVMs) and a lightweight convolutional neural network (CNN) on a pooled TESS+RAVDESS corpus (4,060 utterances, seven emotions). We extract 13-dimensional MFCCs and apply PCA/NMF compression under two settings—2×2 bases (2 PCA + 2 NMF → 26-dim) and 4×4 bases (4 PCA + 4 NMF → 52-dim). An RBF-kernel SVM trained on reconstructed MFCC-statistic vectors attains only 48.2 % test accuracy at 2×2 bases and 64.97 % at 4×4 bases, highlighting its sensitivity to aggressive feature reduction. In contrast, a 490,967-parameter CNN (1.87 MB) trained on the same 2×2 inputs achieves 92 % accuracy and robust per-emotion F scores (0.90–0.94). A learning-curve analysis—varying train size from 10 % to 100 % in 5 % increments—shows that although the CNN initially underperforms the SVM at 10 % data, it quickly overtakes and maintains superior accuracy thereafter, reaching 90 % with full data. These findings elucidate the trade-off between memory footprint and classification performance, suggesting that highly constrained devices may favor compact SVM pipelines, whereas scenarios with sufficient storage and data should leverage CNNs for their superior scalability and accuracy.**

***Index Terms*—Emotion Classification, MFCC features, Principal Component Analysis, Non-Negative Matrix Factorization, Support Vector Machine, CNN**

I. Introduction

Speech emotion recognition (SER) enables machines to interpret human affective states from audio, with applications in virtual assistants, call-center analytics, and affective computing [1], [2]. Traditional ML pipelines—using hand-crafted MFCC features and classifiers like SVMs—offer efficiency and interpretability for resource-constrained settings [2], [5], whereas DL models (e.g., CNNs) learn spectro-temporal patterns automatically and often achieve higher accuracy at the cost of increased data and compute requirements [3]. To bridge this gap, we apply PCA and NMF to compress 13-dimensional MFCCs, reducing dimensionality while preserving salient information [6].

In this study, we pool TESS [7] and RAVDESS [8] into a unified dataset and compare an RBF-kernel SVM (via LOOCV and an 80:20 split) against a lightweight CNN (70:15:15 split), evaluating (i) the effect of feature-basis count on accuracy and (ii) performance scaling as training data grows from 10 % to 100 %. Our results show that, even under aggressive MFCC compression, CNNs consistently outperform SVMs across all data regimes, while SVMs retain advantages in simplicity and stability with very limited components.

The rest of the paper is outlined as follows: in Section II , we review related work. Section III, we explain the background of the SVM,CNN, feature extraction and compress methods; in Section IV, we describe the methodology and how we conducted the experiments; Section V presents the results discussion; and in Section VI, we conclude the study.

II. Related Work

Recent advances in speech emotion recognition (SER) emphasize the critical role of feature extraction methodologies. Sharma et al. [4] comprehensive survey (2019) identifies five key audio feature domains: temporal, frequency, cepstral, wavelet, and time-frequency, with MFCC, PLP, and LPC emerging as dominant features for emotion classification. Their analysis establishes that feature engineering remains fundamental to machine learning performance, particularly for paralinguistic pattern recognition.

In our previous study [9], we compared feature extraction techniques for speech emotion recognition (SER) on the TESS dataset using an RBF-kernel SVM classifier. We evaluated Empirical Mode Decomposition (EMD), Continuous Wavelet Transform (CWT), Wavelet Packet Transform (WPT), and MFCC-based features. While EMD and CWT individually yielded limited performance and their combination achieved 88% accuracy, WPT (32 subbands×3 statistical features) reached 92%, and MFCC+SVM delivered the highest result at 97% accuracy under a 70:30 LOOCV/train–test split, which is indications of possible overfitting because limited variation of the data, i.e, TESS dataset cosists of two actresses and using same phrases "say the word __" in different emotions.

Shah et al. [10] analyzed both traditional ML and DL techniques for SER in psychotherapy applications by pooling RAVDESS, TESS, and SAVEE into a single corpus. Their Random Forest and boosting ensemble classifiers achieved

86.3% and 85.8% overall accuracy, respectively. In contrast, CNN and LSTM models on the same data attained around 75% accuracy, highlighting the trade-off between model complexity and generalization in clinical settings.

Pratama and Sihwi [11] developed an SVM-based SER model using MFCC features extracted from RAVDESS and TESS. They experimented with linear, polynomial, and RBF kernels and varied regularization (C=10,100,1000) across 10-fold stratified splits. Their SVM consistently exceeded 70% accuracy on individual and combined test sets, demonstrating the robustness of MFCC–SVM pipelines for cross-corpus SER.

Islam et al. [12] assessed deep convolutional neural networks on the TESS corpus, augmented with voice variations and white noise. Their CNN outperformed achieving an average recognition accuracy of 98%, underscoring the potential of deep architectures for real-time emotion recognition in high-stakes applications.

III. Background Theory

A. Support Vector Machine (SVM)

Support Vector Machines (SVMs) are supervised binary classifiers that seek the optimal separating hyperplane

$$f(\mathbf{x}) = \mathbf{w}^\top \mathbf{x} + b$$

between two classes $y_i \in \{-1, +1\}$ by maximizing the margin—the distance between the hyperplane and the nearest training points (support vectors) [13]. In the linearly separable case, SVM solves:

$$\min_{\mathbf{w},b} \frac{1}{2}\|\mathbf{w}\|^2 \quad \text{s.t.} \quad y_i(\mathbf{w}^\top \mathbf{x}_i + b) \geq 1, \quad i = 1, \ldots, T.$$

When data are not linearly separable, SVM employs kernel functions

$$K(\mathbf{x}, \mathbf{z}) = \langle \phi(\mathbf{x}), \phi(\mathbf{z}) \rangle$$

to map inputs $\mathbf{x}$ into a higher-dimensional feature space $\phi(\mathbf{x})$. This "kernel trick", allows finding a linear separator in the transformed space without explicit computation of $\phi(\cdot)$. Popular kernels include the polynomial $K(\mathbf{x}, \mathbf{z}) = (\mathbf{x}^\top \mathbf{z} + 1)^p$ and the Gaussian RBF $K(\mathbf{x}, \mathbf{z}) = \exp(-\gamma \|\mathbf{x} - \mathbf{z}\|^2)$, both of which satisfy Mercer's condition and ensure translation invariance for robust classification [13].

B. Convolutional Neural Network (CNN)

Convolutional Neural Networks (CNNs) are a specialized class of deep feedforward neural networks tailored for grid-structured data such as images [14]. A typical CNN architecture alternates:

- **Convolutional layers**, which apply learnable kernels over local receptive fields to produce activation maps. Parameter sharing (the same kernel weights at every spatial location) and sparse connectivity drastically reduce the number of parameters compared to fully connected layers.
- **Pooling layers**, most often max-pooling with 2×2 windows and stride 2, which downsample spatial dimensions to control overfitting and computational cost.
- **Fully-connected layers**, which integrate the hierarchical features for final classification.

Key design choices include small filter sizes (e.g. 3×3), zero-padding and stride settings to manage output dimensions, and ReLU activations to introduce non-linearity. By stacking multiple convolution–ReLU–pooling blocks, CNNs learn hierarchical representations—from edges and textures in early layers to object parts and high-level concepts in deeper layers. This architectural bias toward locality and translational invariance underpins CNNs' state-of-the-art performance on image recognition tasks, while naturally mitigating overfitting through reduced parameter counts and built-in regularization via pooling and weight sharing.

C. Mel Frequency Cepstral Coefficients

The Mel-Frequency Cepstral Coefficients (MFCCs) are widely utilized in audio signal analysis due to their ability to capture perceptually relevant pitch and frequency information. These features are computed using the mel scale, which maps linear frequency to a perceptual scale defined as:

$$\text{Mel}(f) = 2595 \log_{10}\left(1 + \frac{f}{700}\right) \tag{1}$$

This scale reflects the human auditory system's sensitivity to pitch changes across different frequency ranges, typically spanning from 0 Hz to 20,050 Hz. MFCCs are derived by first applying a triangular band-pass filter bank to the FFT power spectrum of a short audio segment. A common implementation uses 12 such filters. The coefficients are then calculated using the discrete cosine transform (DCT) of the log filter outputs:

$$C_n = \sum_{k=1}^{K} \log(S_k) \cos\left[n(k - 0.5)\frac{\pi}{K}\right], \quad n = 1, 2, \ldots, N \tag{2}$$

where S_k represents the k-th filter bank output, and N is the total number of MFCCs extracted from a typical 20 ms audio frame [**?**].

D. Non-negative Matrix Factorization (NMF)

Non-negative Matrix Factorization (NMF) seeks a parts-based, low-rank approximation of a nonnegative data matrix $V \in R_{\geq 0}^{M \times N}$ by finding two nonnegative factors $W \in R_{\geq 0}^{M \times r}$ and $H \in R_{\geq 0}^{r \times N}$ such that

$$V \approx WH,$$

with $r < \min(M, N)$ [15]. Here, each column of W represents a basis component and each row of H gives the corresponding activation across samples, yielding an additive decomposition

$$V = \sum_{k=1}^{r} W_{:,k}\, H_{k,:}\,.$$

The NMF decomposition is typically obtained by minimizing a suitable divergence (e.g. squared-error or Kullback–Leibler)

between V and WH under the constraint $W, H \geq 0$. Iterative optimization methods—such as alternating nonnegative least squares or projected-gradient algorithms—are used to enforce nonnegativity and find a stationary solution. NMF's interpretability and dimensionality reduction properties make it valuable for feature extraction in signal and image analysis.

E. Principal Component Analysis (PCA)

Principal Component Analysis (PCA) is a linear transform method used to obtain a compact, uncorrelated representation of a high-dimensional data vector $\mathbf{y} \in R^K$ by projecting onto the eigenvectors of its covariance matrix $\Sigma_y = E[(\mathbf{y} - \bar{\mathbf{y}})(\mathbf{y} - \bar{\mathbf{y}})^\top]$ [15]. Writing

$$\mathbf{y} = W\,\mathbf{x},$$

where $W = [\mathbf{w}_1, \ldots, \mathbf{w}_K]$ is an orthonormal basis ($W^\top W = I$), PCA chooses $\mathbf{w}_k$ to be the eigenvectors of Σ_y:

$$\Sigma_y\,\mathbf{w}_k = \lambda_k\,\mathbf{w}_k,$$

with eigenvalues $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_K$. The transformed coefficients $x_k = \mathbf{w}_k^\top \mathbf{y}$ are uncorrelated and satisfy $\mathrm{Var}(x_k) = \lambda_k$. To reduce dimensionality to $L < K$, one retains the first L components—minimizing the mean-squared reconstruction error

$$\epsilon^2 = \sum_{k=L+1}^{K} \lambda_k,$$

while discarding lower-variance directions. PCA thus provides an efficient, redundancy-free representation by selecting the principal directions of maximal variance in the data.

IV. Methods

A. Dataset

1) Toronto Emotional Speech Set (TESS): The Toronto Emotional Speech Set (TESS) [7] comprises 2 800 high-quality WAV recordings from two musically trained, native-English female speakers (ages 26 and 64) with normal hearing. Each speaker utters 200 target words—embedded in the carrier phrase "Say the word ____"—under seven emotion labels (anger, disgust, fear, happiness, pleasant surprise, sadness, and neutral), yielding $200 \times 7 \times 2 = 2\,800$ samples. Files are organized by speaker and emotion, providing a clean, balanced corpus for model training.

2) RAVDESS Dataset: The Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS) [8] includes 1 440 audio-only speech recordings (16 bit, 48 kHz WAV) from 24 professional actors (12 male, 12 female). Each actor performs two neutral-accent statements under eight emotional labels (neutral, calm, happy, sad, angry, fearful, disgust, surprise) at two intensity levels. For our work, we excluded "calm" (not present in TESS) and retained 1 260 utterances across the remaining seven emotions.

3) Data Pooling: To create a unified corpus, we merged TESS and RAVDESS samples. From TESS we used all 2 800 recordings (400 per emotion × 7 emotions). From RAVDESS we included six emotions (happy, sad, angry, fearful, disgust, surprise) with 192 recordings each, plus 96 neutral recordings. In total, the pooled dataset comprises

$$(400 \times 7) \;+\; (6 \times 192) \;+\; 96 \;=\; 4\,060$$

utterances for training and evaluation.

4) Noise Augmentation: To improve model robustness, we injected zero-mean Gaussian noise into the MFCC inputs during CNN training. Concretely, if X_{train} denotes the original feature matrix, we generate a noisy copy

$$X_{\text{train}}^{\text{noisy}} = X_{\text{train}} + 0.01 \times \mathcal{N}(0, 1),$$

where the noise standard deviation is set to 0.01. We then concatenate clean and noisy examples along the batch dimension—doubling the training set—and apply a random permutation to shuffle both X and y before each epoch.

B. Feature Extraction

1) MFCC Extraction: Each audio file is converted into a fixed-length MFCC feature matrix $\mathbf{M} \in R^{13\times 85}$ as follows. Let y be the waveform of length N samples loaded at sampling rate sr. We choose the hop length

$$h = \left\lfloor \frac{N}{T-1} \right\rfloor,$$

where $T = 85$ is the desired number of time frames. We then compute the MFCC matrix

$$\mathbf{M} = \mathrm{MFCC}(y;\, n_{\mathrm{mfcc}} = 13,\; h,\; sr),$$

which yields a $13 \times L$ array with L frames. If $L < T$, we zero-pad along the time axis:

$$\mathbf{M} \leftarrow [\mathbf{M} \;|\; \mathbf{0}_{13\times(T-L)}],$$

otherwise we truncate:

$$\mathbf{M} \leftarrow \mathbf{M}_{:,\,1:T}.$$

This ensures every utterance is represented by a 13×85 MFCC matrix for downstream SVM and CNN models.

Beyond our primary pipeline (26-dim mean+std summaries → SVM vs 13×85 MFCC maps → CNN), we also ran an SVM on the full 13×85 MFCC maps (flattened to 1 105 dimensions) to isolate the effect of classifier architecture. This "full-map" SVM used the same RBF-kernel (`gamma='scale'`, $C = 10$) but operated on the uncompressed time–frequency input, after padding/truncating every sample to 85 frames as in the CNN pipeline.

2) Mel-Spectrogram Extraction: Each audio file is converted into a fixed-size mel-spectrogram of shape $n_{\text{mels}} \times T \times 1$ as follows. Let $y \in R^N$ be the waveform loaded at sampling rate sr. We compute the mel-spectrogram

$$S_{m,l} = \text{MelSpectrogram}(y;\ sr,\ n_{\text{mels}}),$$

where, $m = 1, \ldots, n_{\text{mels}},\ l = 1, \ldots, L,$

yielding $S \in R^{n_{\text{mels}} \times L}$. We then convert to log-power decibels,

$$\tilde{S}_{m,l} = 10 \log_{10}\Big(\frac{S_{m,l}}{\max_{i,j} S_{i,j}}\Big).$$

To enforce a uniform frame count $T = 85$, we zero-pad or truncate along the time axis:

$$\hat{S}_{m,l} = \begin{cases} \tilde{S}_{m,l}, & 1 \le l \le \min(L, T), \\ 0, & L < l \le T. \end{cases}$$

Finally, we add a singleton channel dimension so that $\hat{\mathbf{S}} \in R^{n_{\text{mels}} \times T \times 1}$, making it compatible with our 2D CNN input.

C. Model Architectures

1) SVM Training: We train a support vector machine with a radial-basis-function (RBF) kernel

$$K(\mathbf{x}, \mathbf{z}) = \exp\big(-\gamma \|\mathbf{x} - \mathbf{z}\|^2\big),$$

where γ is set to $1/d$ (with d the feature dimension) and the regularization parameter $C = 10$. The classifier is obtained by solving the convex dual problem

$$\max_{\{\alpha_i\}} \sum_{i=1}^{N} \alpha_i \ - \ \tfrac{1}{2} \sum_{i,j=1}^{N} \alpha_i \alpha_j y_i y_j \, K(\mathbf{x}_i, \mathbf{x}_j)$$

subject to

$$0 \le \alpha_i \le C, \quad \sum_{i=1}^{N} \alpha_i y_i = 0.$$

The resulting decision function is

$$f(\mathbf{x}) \ = \ \text{sign}\Big(\sum_{i=1}^{N} \alpha_i \, y_i \, K(\mathbf{x}_i, \mathbf{x}) + b\Big),$$

which we fit on the training set (either via leave-one-out cross-validation or an 80:20 split) and then apply to predict labels on held-out data.

D. Evaluation Metrics

Performance is assessed using a comprehensive set of evaluation metrics to quantify the effectiveness of the classification model. These include accuracy, precision, recall, and F1-score, computed for each emotion class. Additionally, confusion matrices are plotted to provide insight into misclassification patterns across emotional categories.

- **Accuracy:** Accuracy measures the proportion of correctly classified samples out of the total number of samples. It is defined as:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

where TP, TN, FP, and FN represent the number of true positives, true negatives, false positives, and false negatives, respectively. While useful for general performance, accuracy may be misleading in imbalanced datasets.

- **Precision:** Precision quantifies the proportion of correctly predicted positive instances among all predicted positives for a given class. It is given by:

$$\text{Precision} = \frac{TP}{TP + FP}$$

A high precision indicates a low false positive rate, which is important in applications where false alarms are costly.

- **Recall:** Also known as sensitivity or true positive rate, recall measures the ability of the model to correctly identify all actual positive instances of a class:

$$\text{Recall} = \frac{TP}{TP + FN}$$

High recall is crucial when missing true instances (false negatives) is more critical than falsely identifying them.

- **F1-Score:** The F1-score is the harmonic mean of precision and recall, offering a balanced measure when there is an uneven class distribution:

$$\text{F1} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

It is particularly useful when both false positives and false negatives carry significant consequences.

- **Confusion Matrix:** A confusion matrix provides a tabular representation of the predicted vs. actual classifications, allowing visualization of correct and incorrect classifications for each class. The diagonal elements represent correctly classified instances, while off-diagonal elements indicate misclassifications. Normalized confusion matrices (i.e., values represented as proportions) are used to account for class imbalance and better interpret the model's behavior.

V. Experiment and Results

For our Experiments, we have used tensorflow V 2.10 and sci-kit 1.16. The Training was done on Nvidia RTX GPU 3060 GPU using tensorflow, while SVM trained on the CPU.

A. SVM vs. CNN on Combined Data

1) SVM with Noise Augmentation: We constructed 26-dimensional feature vectors by concatenating the per-coefficient mean and standard deviation of 13 MFCCs. The raw training set contained 3238 samples (3238×26), which we doubled to 6476 by adding zero-mean Gaussian noise ($\sigma = 0.01$). After standardizing the augmented set, we performed leave-one-out cross-validation (LOOCV), achieving an overall accuracy of 86% in 7161.5s of training. Table I summarizes the LOOCV precision, recall and F1-scores by emotion.

TABLE I
LOOCV RESULTS ON NOISE-AUGMENTED TRAINING SET (6476 SAMPLES)

Emotion	Precision	Recall	F1-score
angry	0.88	0.87	0.87
disgust	0.88	0.86	0.87
fear	0.93	0.86	0.89
happy	0.87	0.86	0.87
neutral	0.97	0.86	0.91
pleasant_surprised	0.68	0.89	0.77
sad	0.91	0.84	0.87
LOOCV accuracy: 0.86; training time: 7161.5s			

Finally, we trained a single RBF-kernel SVM on the full augmented set and evaluated it on an 810-sample test split. This yielded a test accuracy of 72% (training time: 1.09s). Table II reports the per-emotion F1-scores on the held-out test set, and Figure 1 displays the normalized confusion matrix.

TABLE II
TEST-SET RESULTS ON AUGMENTED SVM

Emotion	Precision	Recall	F1-score
angry	0.75	0.73	0.74
disgust	0.77	0.75	0.76
fear	0.81	0.72	0.76
happy	0.78	0.71	0.74
neutral	0.82	0.76	0.79
pleasant_surprised	0.51	0.73	0.60
sad	0.76	0.68	0.72
Test accuracy: 0.72; final training time: 1.09s			

When the SVM received the full 13×85 MFCC map (flattened to 1 105 features) instead of the 26-dim summary vector, its test accuracy rose from 73 % to 93 %. Two factors explain this jump:

- Richer input. The full map preserves all fine-grained temporal patterns in the MFCCs, whereas mean+std discards most frame-level detail.
- Kernel scaling. With gamma='scale' (i.e. $\gamma = 1/(d \cdot \mathrm{Var}(X))$), increasing d from 26 to 1,105 automatically reduces γ by ∼42×, producing a smoother decision boundary that often generalizes better.

2) CNN on Pooling Dataset: The EmotionClassifier Fig. **??** is a compact convolutional neural network of approximately 1.14 million parameters (4.36 MB) that transforms a 64×64 input into one of seven emotion categories. It begins with three convolutional blocks—each comprising a 3×3 convolution with L2 weight decay followed by batch normalization, ReLU activation, 2×2 max-pooling and 30% dropout—where the number of filters doubles from 32 to 64 to 128 while the spatial resolution is halved from 64×64 down to 8×8. The resulting 8×8×128 feature map is flattened into a vector of 8,192 units and passed through a 128-unit dense layer with ReLU activation and 40% dropout, before a final 7-unit softmax layer produces the class probabilities. Throughout the network, weight decay, batch-norm non-trainable statistics and multiple dropout layers work together to regularize learning and guard against overfitting, making the model well suited to emotion-recognition tasks on moderate-sized datasets.

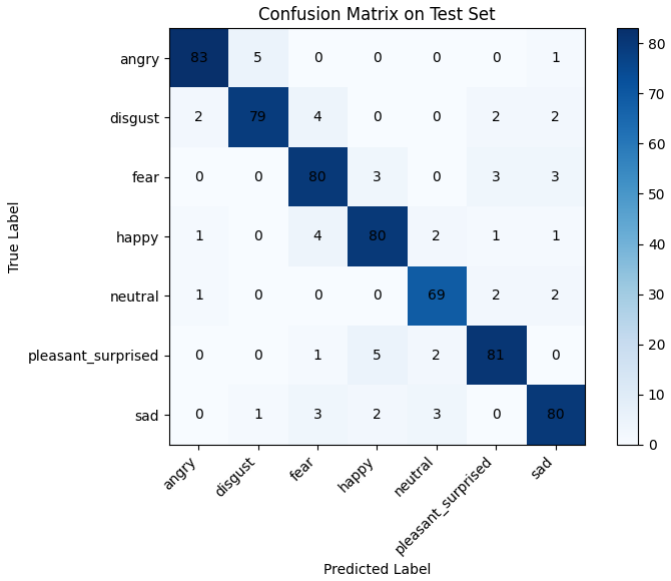

Fig. 1. CNN test-set confusion matrix.

Table III summarizes the data shapes and augmentation for the CNN experiment. After splitting and augmenting, the network was trained for 50 epochs.

TABLE III
CNN INPUT AND AUGMENTATION SHAPES

Split	**Shape**
Train	(2832, 13, 85, 1)
Val	(608, 13, 85, 1)
Test	(608, 13, 85, 1)
Augmented train	(5664, 13, 85, 1)

On the held-out test set, the CNN achieved an overall accuracy of 90.8%. Table IV details per-emotion precision, recall, and F1 scores.

TABLE IV
PER-EMOTION TEST-SET METRICS FOR CNN

Emotion	Precision	Recall	F1-score
angry	0.933	0.944	0.939
disgust	0.890	0.910	0.900
fear	0.891	0.921	0.906
happy	0.879	0.899	0.889
neutral	0.972	0.932	0.952
pleasant_surprised	0.866	0.944	0.903
sad	0.961	0.820	0.885

B. Spectrogram-CNN Performance

We evaluated a deeper CNN on 64-channel Mel-spectrogram inputs of shape $(64 \times 85 \times 1)$. After noise augmentation (doubling the 2 832 raw training samples to 5 664), the network—with six convolutional blocks and two dense layers—totaled approximately 4.2 million parameters.

Training over 50 epochs required 160.8 s in total (5.4 s/epoch). Validation accuracy climbed from 24.3 % in epoch 1 to 78.1 % by epoch 7 and plateaued in the low-90s. On the held-out test set, the model achieved a test loss of 0.892 and an overall accuracy of 86.0 %.

Table V reports per-emotion precision, recall, and F1-score on the test set.

TABLE V
PER-EMOTION TEST-SET METRICS FOR SPECTROGRAM-CNN

Emotion	Precision	Recall	F1-score
angry	0.929	0.876	0.902
disgust	0.938	0.854	0.894
fear	0.735	0.933	0.822
happy	0.864	0.787	0.824
neutral	0.983	0.797	0.881
pleasant_surprised	0.987	0.876	0.929
sad	0.718	0.888	0.794

C. Feature Reduction Performance

1) Feature-Basis Analysis: We applied PCA to the 13-dimensional MFCC feature vectors for three conditions—TESS only, RAVDESS only, and the pooled corpus—and plotted the first two principal components to compare the emotion centroids. Across all three projections, *Sad* and *Neutral* lie together on the far right of PC1, indicating highly similar spectral profiles. The overall circular ordering of the seven emotions (when traversing clockwise or counter-clockwise) is almost preserved in the separate TESS and RAVDESS plots Fig2, though the merged corpus slightly distorts that sequence. *Disgust* clusters near the origin along PC1 in the individual datasets but shifts upward in the combined projection, suggesting a corpus-interaction effect. By contrast, *Happiness* consistently appears in the upper-left quadrant—furthest from the center—reflecting its distinctive high-energy envelope. *Anger* remains on the left side in all cases, though in TESS it sits mid-height on PC2, whereas in RAVDESS and the pooled data it falls lower. *Pleasant surprise* lies low in both TESS and RAVDESS but moves to the upper-middle region when the corpora are merged. Finally, *Fear* does not form a tight cluster in any plot, yet it always occupies the negative PC1 half—consistent with its intermediate spectral signature. Notably, in TESS (and to some extent the combined set) each emotion also splits into two sub-clusters symmetrically arranged around its centroid, which may correspond to speaker-specific variations (e.g. the two actresses).

Complementing the PCA results, we performed NMF on the same MFCC matrices and examined the dominant components for each emotion. In RAVDESS, the first NMF basis overwhelmingly captures the spectral variance—its coefficient ranges from 0.8 to 1.3—while all subsequent bases remain comparatively small. In TESS, *Neutral* and *Sad* attain the highest weight on component 1 (1.3), followed by *Disgust* and *Fear* on components 2 (0.65) and 3 (0.53), with the other

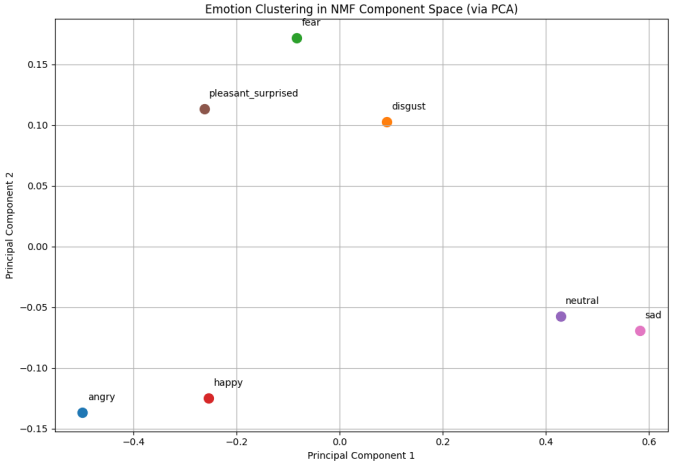

Fig. 2. PCA of the seven emotion centroids of the pooling data

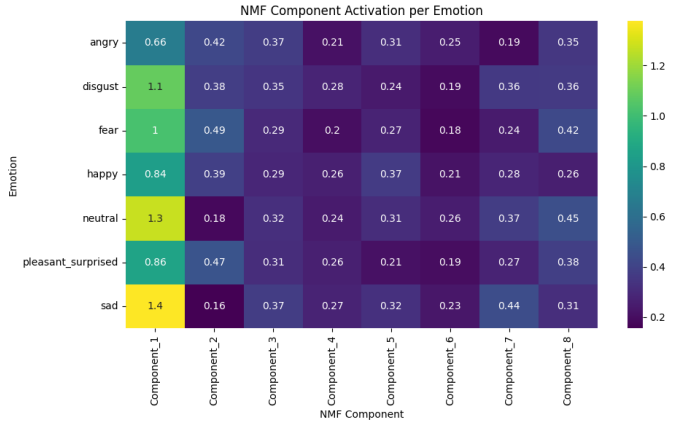

Fig. 3. NFM of the seven emotion of the pooling data

emotions scoring below 0.4—except *Happiness*, which peaks around 0.5 on component 5, and *Pleasant surprise*, which shows a modest rise on components 2–3. When TESS and RAVDESS are pooled, Fig. 3, both the first and second NMF bases carry the largest weights, but component 1 remains substantially larger, indicating that a single "global" spectral part still dominates the combined dataset. These findings demonstrate that although one or two additive spectral bases explain most of the variance, their relative importance and order shift upon merging the two corpora.

2) Results After Features Reduction: We evaluated how extreme compression of the MFCC feature space affects both our CNN and SVM classifiers by retaining only the top PCA and NMF components. Two compression settings were tested: *2×2 bases* (2 PCA + 2 NMF components, yielding 26 input features) and *4×4 bases* (4 PCA + 4 NMF, yielding 52 features).

a) SVM Results: Using the same 2×2 (NMFxNMF and PCAxPCA) compressed features, an RBF-kernel SVM attained only **51.15 %** validation accuracy and **48.19 %** test accuracy. When we increased to 4×4 bases, SVM performance

rose to **67.11 %** on validation and **64.97 %** on test, but remained substantially below the CNN. This contrast highlights the SVM's sensitivity to aggressive feature reduction and its need for more retained components to recover acceptable accuracy.

b) CNN Results : Our lightweight CNN (490,967 parameters, 490,967 trainable; model size 1.87 MB) trained on the 2×2 compressed inputs achieved **92 %** test accuracy on the 15% held-out utterances. Its per-emotion F1-scores ranged from 0.90 to 0.94 (e.g., angry 0.94, neutral 0.94, pleasant_surprised 0.93), demonstrating that the deep, convolutional architecture remains highly effective even after reduction to almost 40% in size of the model and used features.

TABLE VI
IMPACT OF MFCC FEATURE COMPRESSION ON SVM AND CNN PERFORMANCE

Model	Compression	# Feat.	Test Acc.
SVM	2×2 (2 PCA+2 NMF)	26	48.19 %
SVM	4×4 (4 PCA+4 NMF)	52	64.97 %
CNN	2×2 (2 PCA+2 NMF)	26	92 %

D. Learning-Curve Analysis

To assess and compare the data-efficiency of the classical ML approach (SVM on MFCC features) versus the deep learning approach (CNN on MFCC-map inputs), we conducted a systematic learning-curve analysis. We varied the fraction of available training data from 10 % up to 100 % in 5 % increments, training each model from scratch on the corresponding subset while evaluating performance on a held-out test set. All experiments were repeated with a fixed random seed (SEED = 42) to ensure reproducibility.

For the ML baseline, we extracted 13 MFCC coefficients per frame from TESS and RAVDESS utterances (sampled at 8 kHz and truncated/padded to 0.5 s), computing the mean and standard deviation of each coefficient as a 26-dimensional feature vector. These were standardized via StandardScaler before training an RBF-kernel SVM (C = 10, ='scale'). For the DL model, we generated 13×85 MFCC-maps at 16 kHz, zero-padded or truncated to length 85, and trained a small CNN (five convolutional blocks with batch normalization and ReLU, followed by dense output layers) for 50 epochs with the Adam optimizer (batch size = 32). Both methods used an 80/20 stratified train/test split.

As shown in Figure 4, the SVM baseline achieves moderate accuracy even with limited data: roughly 72 % at 10 % of the training set, plateauing near 75 % beyond 60 % of the data. In contrast, the CNN starts lower (60 % at 10 %) but exhibits a steeper upward trajectory, surpassing the SVM around 50 % training data and reaching 90 % accuracy when trained on the full dataset. This indicates that while handcrafted MFCC-statistics suffice for small-data regimes, the CNN's capacity to learn hierarchical spectro-temporal features becomes advantageous as more labeled samples become available.

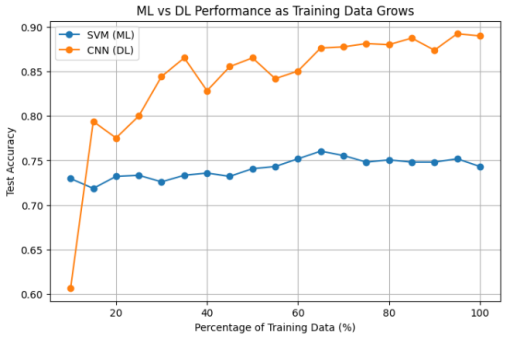

Fig. 4. Graph demonstrating the performance (ML Vs DL) depending on size of data

The SVM's relatively flat curve suggests a low-variance, high-bias regime: its performance gains marginally with additional data, implying that the fixed feature representation and kernel limit its expressivity. By contrast, the CNN demonstrates higher variance (evidenced by lower initial accuracy) but lower bias when trained on larger datasets, capitalizing on end-to-end feature learning to model complex patterns. These observations align with the bias–variance trade-off: ML methods with strong regularization can be more stable with scarce data, whereas DL methods demand ample data to realize their expressive potential.

VI. DISCUSSION

A. Trade-off Between Accuracy and Memory

Extreme compression of the MFCC feature space directly reduces both the storage required for input vectors and the overall model footprint, but yields markedly different impacts on classification accuracy for SVM versus CNN. With only 2×2 bases (2 PCA + 2 NMF, 26 floats 104 bytes per sample), the SVM's validation accuracy falls to 51.15 % and test accuracy to 48.19 %, recovering to 64.97 % (test) only when doubled to 4×4 bases (52 floats 208 bytes) with a modest increase in feature-storage. By contrast, our CNN—although orders of magnitude larger in memory (490,967 parameters 1.87 MB)—maintains 92 % test accuracy on the 2×2 compressed inputs.

These results highlight a clear design trade-off:

- **Memory-constrained scenarios:** If total model size must be kept to the low-kilobyte range, an aggressively compressed SVM (2×2 bases) affords minimal footprint at the cost of sub-50 % accuracy. Increasing to 4×4 bases doubles feature storage yet still limits accuracy to 65 %.
- **Accuracy-critical deployments:** Allocating 2 MB for the CNN yields near-state-of-the-art performance even under aggressive feature reduction—and could be further reduced via quantization or pruning without substantial accuracy loss.

In practice, one must balance the acceptable accuracy threshold against available memory: small-footprint SVMs may suit embedded or low-power devices when moderate performance

suffices, whereas CNNs become preferable when higher accuracy justifies their larger memory footprint.

B. Limitations and Future Directions

While our comparative study highlights the strengths of both SVM and CNN approaches for speech emotion recognition, several limitations must be acknowledged. First, although MFCC-based SVMs are lightweight and fast to train, their reliance on hand-crafted statistical summaries inherently discards temporal and fine-grained spectro-temporal patterns, limiting ultimate accuracy and robustness—particularly when faced with naturalistic, noisy recordings or cross-corpus variability. Second, our CNN experiments, despite demonstrating strong performance even under aggressive MFCC compression, still require a relatively large number of labeled examples (hundreds to thousands) and substantial compute for training. This creates a barrier for applications with scarce annotations or strict latency/power constraints (e.g., wearable devices or real-time edge inference).

Moreover, our analysis focused on clean, balanced corpora (TESS+RAVDESS) with acted emotions; real-world data often exhibit imbalanced class distributions, spontaneous expressions, and background noise, which may degrade model generalization. We also did not evaluate cross-speaker or cross-language transfer performance, leaving open questions about domain shift and the need for adaptation techniques.

Looking ahead, several avenues could strengthen and extend this work:

- **Data Augmentation and Synthetic Generation.** Beyond Gaussian noise, advanced augmentation strategies—such as vocal tract length perturbation, pitch shifting, and adversarial example synthesis—could enrich underrepresented emotion classes and improve robustness to channel and recording variations.
- **Model Compression and Acceleration.** To reconcile the accuracy–memory trade-off, techniques such as weight pruning, low-bit quantization, knowledge distillation, and neural architecture search for lightweight backbones can yield sub-megabyte CNNs without substantial loss of accuracy, enabling deployment on microcontrollers or smartphones.
- **Domain Adaptation and Generalization.** Incorporating unsupervised domain adaptation methods—such as adversarial training or discrepancy-based regularization—can help align feature distributions across different speakers, languages, or recording conditions, enhancing model robustness in the wild.

By pursuing these directions, we aim to develop more data-efficient, robust, and compact emotion recognition models that bridge the gap between laboratory performance and real-world applicability.

VII. Conclusion

In this paper, we systematically compared classical SVM and lightweight CNN classifiers for speech emotion recognition on a pooled TESS+RAVDESS corpus under matched conditions and aggressive MFCC feature-space compression via PCA and NMF. Our experiments demonstrate that SVMs trained on 26-dimensional reconstructed MFCC–statistic vectors suffer significant accuracy degradation under extreme compression (48.2 % test accuracy at 2×2 bases), recovering only to around 65 % when expanded to 4×4 bases, whereas a 1.87 MB CNN trained on the same compressed inputs achieves 92 % test accuracy with robust per-emotion F_1 scores of 0.90–0.94. Learning-curve analysis reveals that although the CNN initially underperforms the SVM when trained on only 10 % of the data, it rapidly overtakes the SVM as more data are added and maintains superior accuracy thereafter, ultimately reaching approximately 90 % accuracy on the full dataset. These results highlight a clear trade-off between memory footprint and classification performance: compressed SVM pipelines are advantageous in highly constrained environments when moderate accuracy suffices, while CNNs become the preferable choice when a few megabytes of storage and sufficient data justify their superior performance. Future work will explore semi-supervised and self-supervised pretraining to boost CNN accuracy in low-data settings, model-compression techniques (quantization, pruning) to narrow the memory gap, and extension of this framework to other paralinguistic tasks to validate its broader utility.

References

[1] M. Płaza *et al.*, "Emotion Recognition Method for Call/Contact Centre Systems," *Applied Sciences*, vol. 12, no. 21, Art. 10951, 2022.

[2] M. El Ayadi, M. S. Kamel, and F. Karray, "Survey on speech emotion recognition: Features, classification schemes, and databases," *Pattern Recognition*, vol. 44, no. 3, pp. 572–587, 2011.

[3] J. Zhao, X. Mao, and L. Chen, "Speech emotion recognition using deep 1D & 2D CNN LSTM networks," *Biomedical Signal Processing and Control*, vol. 47, pp. 312–323, 2019.

[4] G. Sharma, K. Umapathy, and S. Krishnan, "Trends in audio signal feature extraction methods," *Applied Acoustics*, vol. 158, p. 107020, 2020, doi: 10.1016/j.apacoust.2019.107020.

[5] P. Partila, M. Voznak, and J. Tovarek, "Pattern Recognition Methods and Features Selection for Speech Emotion Recognition System," *The Scientific World Journal*, vol. 2015, Art. ID 573068, 2015.

[6] S. R. Bandela and T. K. Kumar, "Speech emotion recognition using semi-NMF feature optimization," *Turkish Journal of Electrical Engineering and Computer Sciences*, vol. 27, no. 5, pp. 3741–3757, 2019.

[7] M. K. Pichora-Fuller and K. Dupuis, "Toronto Emotional Speech Set (TESS) [Dataset]," Borealis Dataverse (Scholars Portal), 2020. DOI: 10.5683/SP2/E8H2MF.

[8] S. R. Livingstone and F. A. Russo, "The Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS): A dynamic, multimodal set of facial and vocal expressions in North American English," *PLOS ONE*, vol. 13, no. 5, Art. e0196391, 2018.

[9] O. Harmouche, "Optimizing Feature Extraction for Speech Emotion Recognition: A Comparative Study of Wavelet, EMD, and MFCC Performance," M.S. thesis, Dept. of Electrical, Computer and Biomedical Engineering, Toronto Metropolitan University, Toronto, Canada, 2024.

[10] N. Shah, K. Sood, and J. Arora, "Speech emotion recognition for psychotherapy: An analysis of traditional machine learning and deep learning techniques," in *Proc. 13th Annual Computing and Communication Workshop and Conference (CCWC)*, 2023, pp. 20–27.

[11] A. Pratama and S. W. Sihwi, "Speech emotion recognition model using support vector machine through MFCC audio feature," in *Proc. 14th Int. Conf. Information Technology and Electrical Engineering (ICITEE)*, Yogyakarta, Indonesia, 2022, pp. 303–308, doi:10.1109/ICITEE56407.2022.9954111.

[12] M. M. Islam, M. A. Kabir, A. Sheikh, M. S. Saiduzzaman, A. Hafid, and S. Abdullah, "Enhancing speech emotion recognition using deep convolutional neural networks," in *Proc. 9th Int. Conf. Machine Learning Technologies (ICMLT)*, Oslo, Norway, May 2024, pp. 1–6, doi:10.1145/3674029.3674045.

[13] S. Krishnan, *Biomedical Signal Analysis for Connected Healthcare*, Elsevier, 2021.

[14] K. O'Shea and R. Nash, "An Introduction to Convolutional Neural Networks," arXiv:1511.08458 [cs.NE], Dec. 2015.

[15] R. M. Rangayyan and S. Krishnan, *Biomedical Signal Analysis*, 3rd ed., IEEE Press Series in Biomedical Engineering, IEEE Press, Canada, 2018.