Supplementary Information (SI) for Sustainable Food Technology.
This journal is © The Royal Society of Chemistry 2025

Supplementary data

Advanced Machine Learning Techniques for Hyacinth Bean Identification through Infrared Spectroscopy and Computer Vision

Pratik Madhukar Gorde, Poonam Singha and Sushil Kumar Singh*

Department of Food Process Engineering, National Institute of Technology Rourkela, Odisha-769008, India

*Corresponding Author:

Dr. Sushil Kumar Singh, Email: singhsk@nitrkl.ac.in; sksingh32325@gmail.com

Table S1. Detailed overview of models and algorithms used in hyacinth bean identification

Models and algorithms	Sub-models	Working principle	Key features	Advantages	Limitations
Regression models	Linear regression	Model's linear relationships between spectral features and chemical properties.	Simple and interpretable	Computationally efficient, works well with small datasets	Poor performance on non-linear data
	Decision tree regression	Splits the dataset into branches using decision rules.	Handles non-linearity well	Easy to interpret, non-parametric	Prone to overfitting
	Random forest regression	Uses multiple decision trees and averages their outputs.	Reduces variance compared to single trees	Handles large datasets well	Computationally expensive
	Support vector regression	Uses kernels to map data into higher dimensions for better separation.	Effective on small datasets with non-linear trends	Works well for complex relationships	Slow for large datasets
	Gaussian process regression	Models the distribution over functions using a probability approach.	Provides uncertainty estimation	Works well with small data	Computationally intensive
	Neural network regression	Uses layers of neurons to model non-linear relationships in spectral data.	High accuracy with large datasets	Capable of handling complex patterns	Requires large computational power
	Kernel ridge regression	Uses kernel methods to model non-linear relationships.	Effective for complex patterns	Provides better regularization	Computationally expensive
	Gradient boosting regression	Uses boosting technique to combine weak learners into a strong predictor.	Improves accuracy	Reduces bias and variance	Requires careful tuning
Classification models	K-Nearest Neighbors	Assigns class based on the majority vote of k-nearest neighbors.	Non-parametric, simple to implement	Effective for small datasets	Sensitive to noisy and irrelevant features
	Support vector machine	Finds an optimal hyperplane that maximizes class separation.	Works well with high-dimensional data	Robust to overfitting	Slow with large datasets
	Naïve Bayes	Uses Bayes' theorem assuming feature independence.	Fast and efficient	Works well with small data	Assumes independence, which is rarely true
	Decision trees	Splits data based on feature thresholds.	Easy to interpret	Handles categorical & numerical data	Overfits easily
	Random forest	Uses multiple decision trees for voting.	Reduces overfitting	More accurate than single trees	High computational cost
	Ensemble learning (Bagging, Boosting, Adaboost, Xgboost)	Combines multiple weak models to form a strong classifier.	Improves performance	Reduces variance and bias	Requires careful hyperparameter tuning
	Neural networks	Uses deep learning architectures to	Highly flexible and	Can learn complex	Computationally

		identify complex patterns.	accurate	relationships	expensive
Deep learning models for image-based classification	EfficientNet_B3	Optimized CNN architecture for feature extraction and scalability.	Pretrained on ImageNet	High accuracy	High computational cost
	EfficientNet_V2_S	Optimized CNN architecture for scalability and efficiency.	Uses squeeze-and-excitation blocks	Highly parameter-efficient	High computational cost
	ConvNeXt_Tiny	Modernized CNN with optimized convolution layers.	Improved efficiency	Works well on large-scale data	Limited transformer-like properties
	MaxVit_T	Hybrid model combining CNN and Vision Transformers.	Captures long-range dependencies	High accuracy	Computationally expensive
	RegNet_Y_1_6GF	Scalable CNN model for improved accuracy.	Adaptive model size	Balances speed and accuracy	Requires high computational resources
	RegNet_Y_3_2GF	Enhanced version of RegNet optimized for efficiency.	Reduced latency	High accuracy	Requires tuning
	DenseNet169	Uses dense connections between layers for improved gradient flow.	Reduces vanishing gradient issues	Improves feature reuse	Requires large memory
	ShuffleNet_V2_X2_0	Lightweight CNN with efficient group convolutions.	Fast inference speed	Low computational cost	Lower accuracy compared to larger models
	MobileNet_V3_Large	Lightweight CNN optimized for mobile applications.	Depth wise Separable Convolutions for efficiency	Fast inference speed	Lower accuracy than larger models
	RegNet_X_3_2GF	Bottleneck CNN architecture with grouped convolutions.	Efficient feature extraction	Optimized accuracy	Computationally expensive
Feature selection & optimization	ReliefF Algorithm	Identifies the most relevant features in spectral data.	Enhances model interpretability	Improves accuracy	May not capture deep relationships
	Savitzky-Golay Filter	Smooths FTIR spectral data to remove noise.	Retains essential signal details	Improves feature clarity	Requires parameter tuning
	Grid Search, Random Search, Bayesian Optimization	Finds the best hyperparameters for models.	Enhances model accuracy	Automates tuning process	Computationally expensive

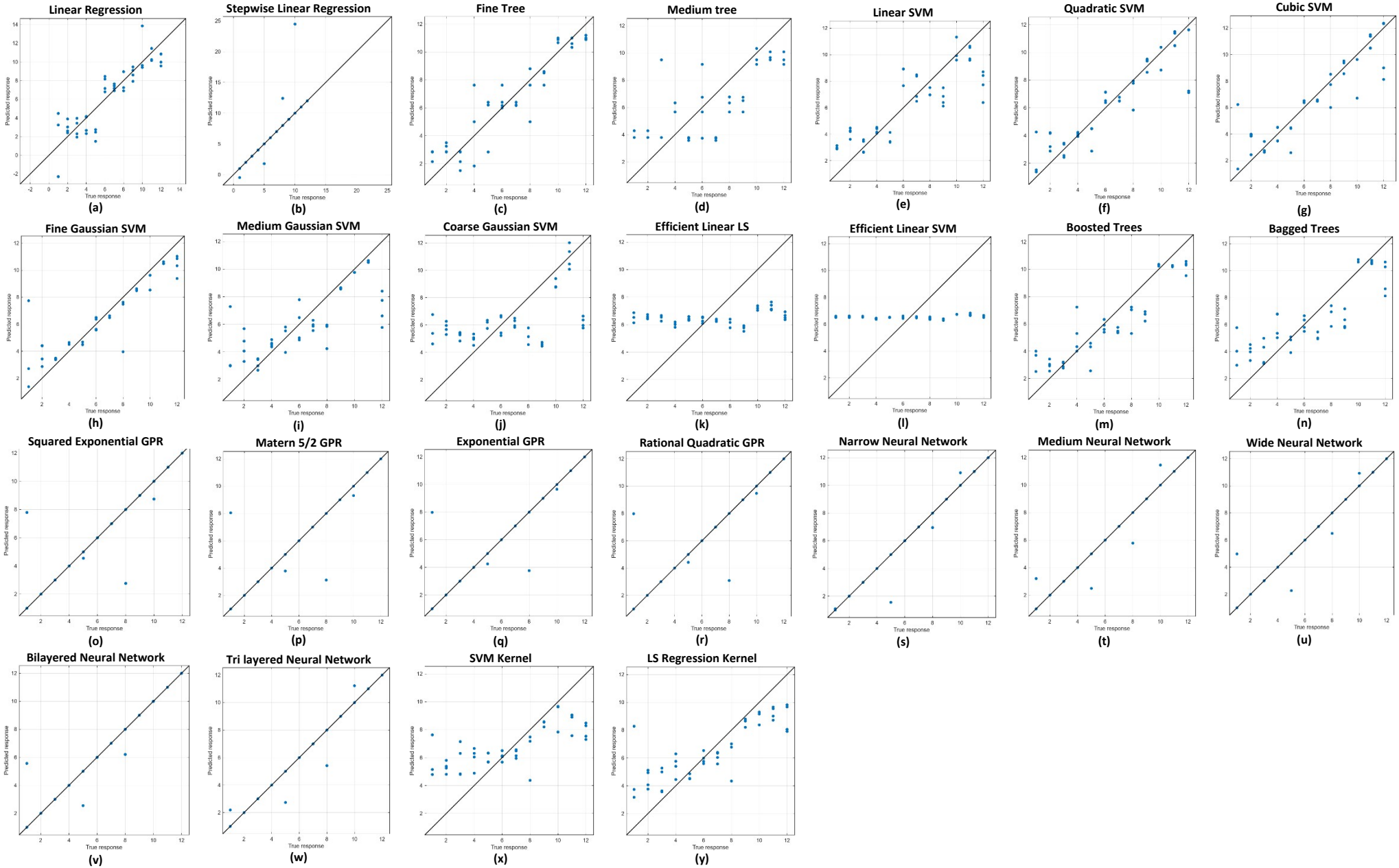

Figure S1. Actual versus predicted value plot for 25 regression models (a-y); SVM- support vector machines; GPR- Gaussian process regression; LS-least square

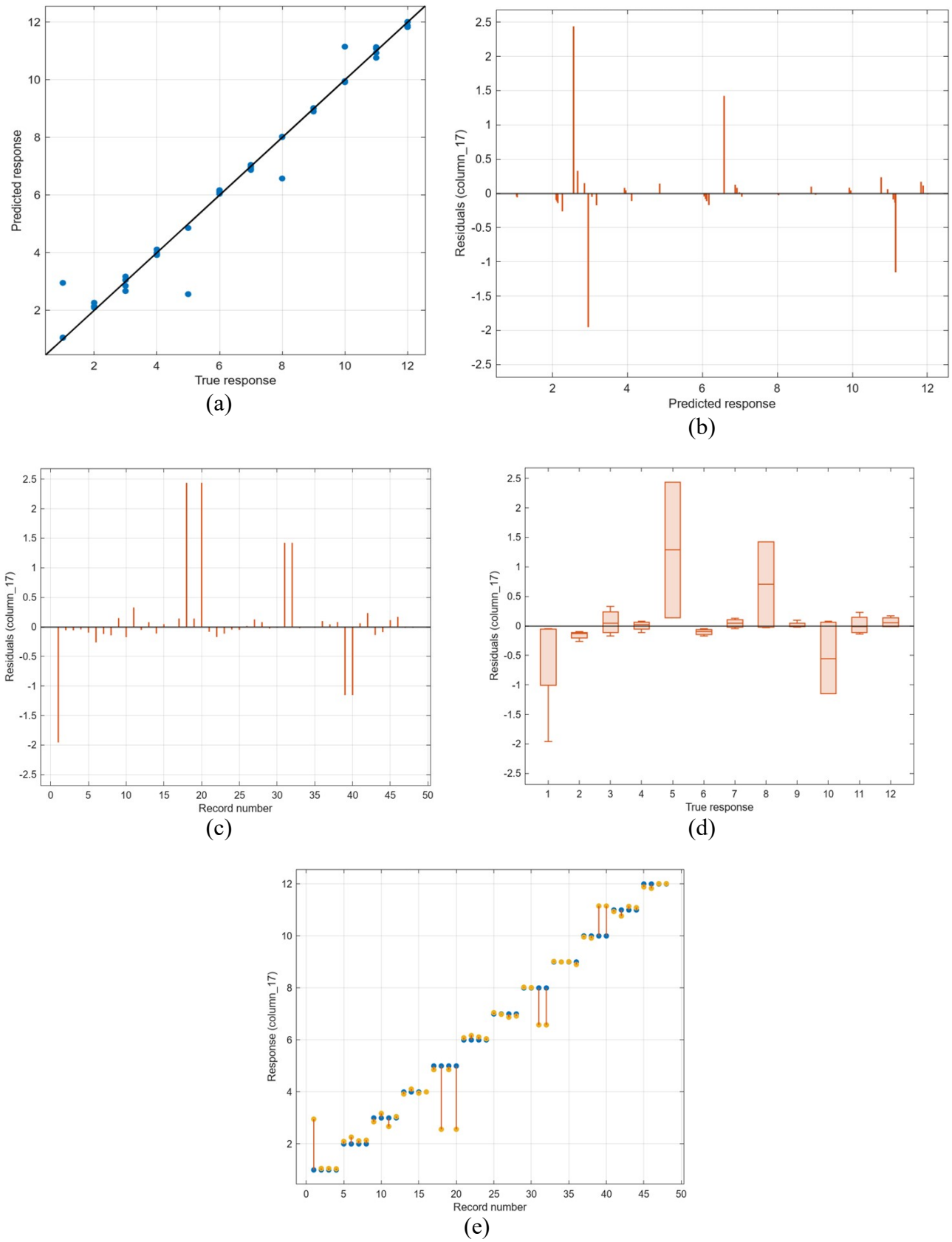

Figure S2. Optimization graphs for neural network model (regression) (a) predicted versus true responses, (b) residuals plot for predicted responses, (c) residuals plot for observations (records), (d) residuals plot for true responses and (e) responses versus observations (records).

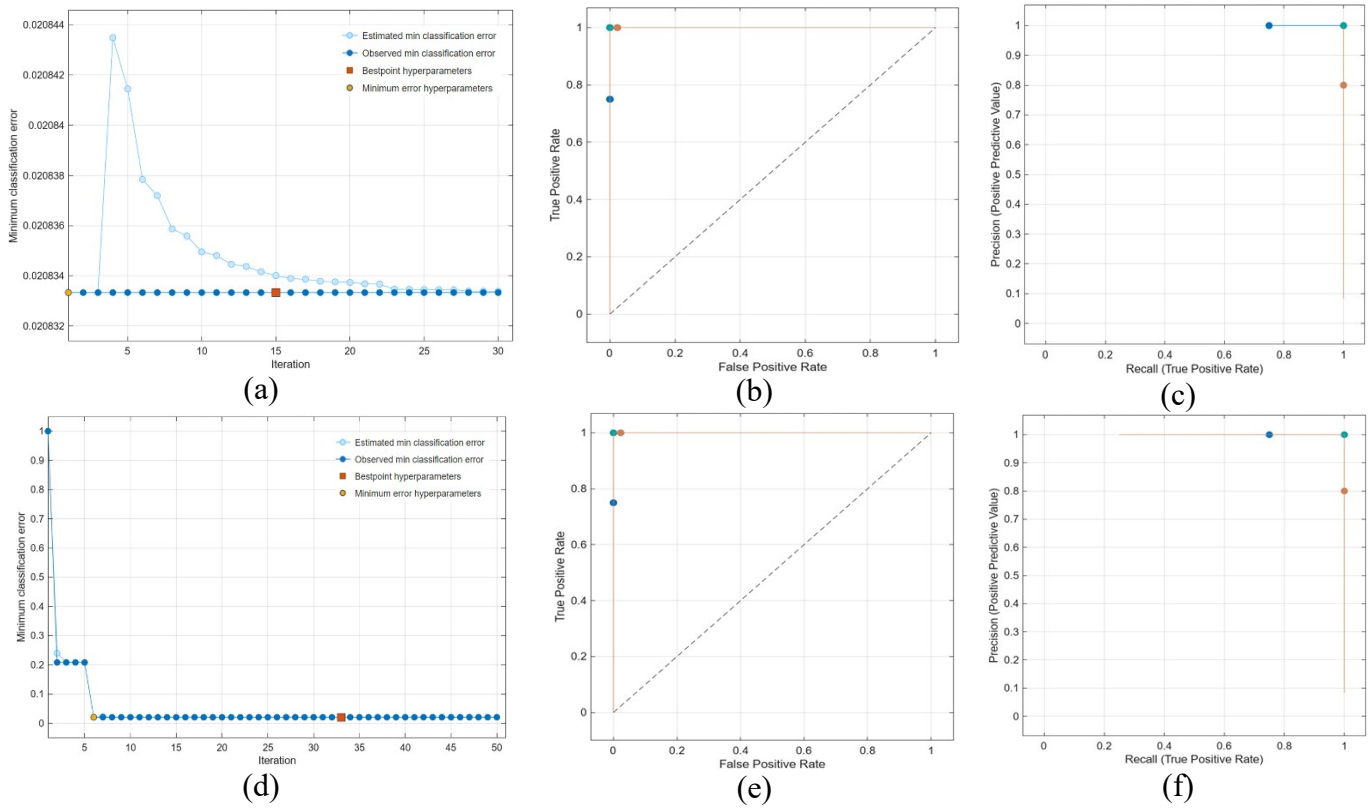

Figure S3. Optimization of classification models; for KNN (a) minimum error classification plot, (b) validation ROC curve, (c) validation precision-recall curve; for NN (d) minimum error classification plot, (e) validation ROC curve, (f) validation precision-recall curve

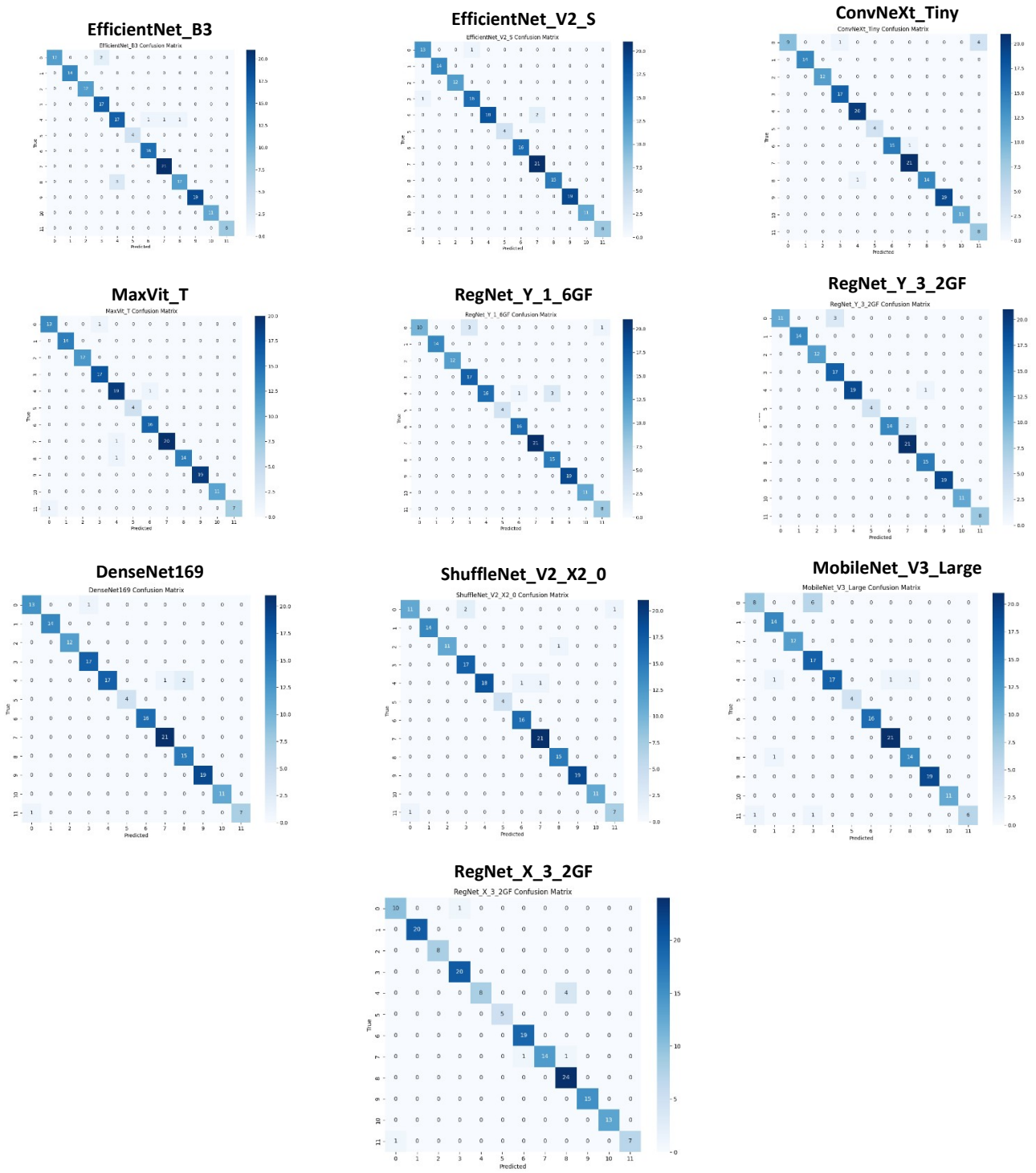

Figure S4. Confusion matrix for selected 10 pre-trained models