Orchard-to-Export: VGG-16 Transfer Learning for Date Fruit Inspection and Quality Grading

Nizamuddin Maitlo[1*], Hidayatullah Shaikh[2], Atique Ur Rehman[3], Basit Raza[4], Fayaz Ahmed[5]

Abstract: **Reliable grading of date fruits is essential for export pricing and compliance but still relies largely on human graders, yielding variable quality, limited throughput, and weak traceability. We propose a compact end-to-end computer-vision pipeline for *variety identification* and *quality grading* using VGG- 16 transfer learning. Experiments use the public *Date Fruit Dataset for Inspection and Grading* (v3) with four varieties (Aseel, Fasli Toto, Gajar, Kupro) organized by size and grade, captured under controlled illumination. Our training recipe applies light augmentation, ImageNet normalization, optional class-balanced sampling, and partial unfreezing of the last VGG block; optimization uses Adam (10^{-4}), batch size 32, early stopping, and cosine-annealing restarts. On a 70/15/15 stratified split, the held-out test set (256 images) yields 98% accuracy with strong per-class performance ($F1 \geq 0.97$ for Aseel and Gajar), with minor confusions between Fasli Toto and Gajar. Learning curves stabilize by epoch 5 without overfitting, and qualitative grids show consistent predictions across sizes and grades. We also outline deployment guidance (illumination control, periodic color calibration, batched real-time inference) and human-in-the-loop verification to support traceability and active learning. Our contributions are an orchard-to-export pipeline, a simple reproducible training recipe for modest datasets, and confusion- aware analyses that surface operational failure modes.**

Keywords: **Date fruit, post-harvest inspection, grading, transfer learning, VGG-16, industrial vision, quality control.**

INTRODUCTION

Dates are a high–value horticultural commodity across the Middle East and South Asia, with growing export flows to Europe and East Asia. Meeting destination-market standards requires consistent recognition of *varieties* and assignment of appearance-based *grades*, tasks traditionally performed by skilled human graders. However, manual visual inspection is slow, fatiguing, and difficult to standardize across shifts and facilities, leading to variability in outcomes and reduced throughput [24]. Recent advances in deep convolutional neural networks (CNNs) have enabled robust, camera-based inspection for fruits and vegetables, particularly under controlled illumination and background conditions typical of post-harvest lines [1], [10]. Within fruit grading, CNNs have displaced handcrafted color/texture pipelines by learning task-relevant features directly from images. Transfer learning from large, generic datasets (e.g., ImageNet) is now the dominant approach when domain data are modest, improving accuracy and convergence while reducing annotation burden [12], [13]. Architectures from the VGG family remain attractive for industrial de- ployments because of their simplicity, stable receptive field growth, and predictable latency on commodity hardware, even as newer backbones achieve higher benchmarks in uncon- strained settings [5]. In the context of date fruits specifically, prior studies report strong results with VGG-based models and related CNNs for variety/quality assessment and real- time inspection, motivating a careful VGG-16 baseline that is practical for small and medium facilities [5], [6], [10].

This work uses the publicly available *Date Fruit Dataset for Inspection and Grading* (v3) containing four varieties Aseel, Fasli Toto, Gajar, and Kupro organized by *size* (Large/Medium/Small) and *grade* (1/2/3). Images were captured under controlled conditions at a fixed camera-to-object distance, reflecting feasible production setups [28]. Leveraging this structure, we formulate a supervised classification problem for variety recognition and quality grading, and we design a training protocol with light geometric/photometric augmenta- tion that remains faithful to production lighting.

Contributions: We make the following contributions:

- ***Orchard-to-export pipeline:*** A compact VGG-16 transfer-learning pipeline for date fruit inspection that unifies variety recognition and quality grading under controlled imaging, with design choices oriented toward deployability.
- ***Simple, reproducible training:*** A pragmatic recipe par- tial unfreezing, mild augmentation, early stopping, and optional class-balanced sampling that stabilizes learning on modest datasets while preserving inference speed [12], [13].
- ***Confusion-aware evaluation:*** Thorough reporting with learning curves, class-wise metrics, confusion matrices, and qualitative grids to surface operational failure modes (e.g., Fasli Toto vs. Gajar look-alikes) [1].
- ***Operational guidance:*** Practical notes for line inte- gration (illumination control, periodic color calibration, batching for throughput) and human-in-the-loop verifica- tion to support traceability and continuous improvement [10], [24].

In comprehensive experiments on a stratified split of the public dataset, our VGG-16 model attains high accuracy with balanced per class performance and stable learning dynamics. We discuss failure modes, ablations, and deployment considerations, and outline extensions to multi-task heads and explainability for production adoption.

[1,2,3,4,5] Institute of Computer Science, Shah Abdul Latif University, Khairpur
Country: Pakistan
Email: *nizamuddin.cs@gmail.com

LITERATURE REVEIW

Classical machine-vision graders in horticulture relied on handcrafted color and texture descriptors (e.g., HSV his- tograms, GLCM, LBP) paired with SVMs or random forests. These pipelines typically began with background subtraction and color normalization, followed by feature computation on the object mask and a shallow classifier. Although competitive in tightly controlled labs, they were brittle to illumination drift, specular highlights, camera replacement, or seasonal changes in surface appearance; moreover, they demanded task-specific feature engineering and frequent re-tuning when the acquisition setup changed. Deep CNNs displaced such pipelines by learning hierarchical features directly from pixels, with transfer learning from large natural-image corpora (e.g., ImageNet) now the de facto strategy when domain datasets are modest or imbalanced [25], [26]. Among standard backbones, VGG-16 remains attractive for industrial deployment thanks to its uniform 3×3 design, stable receptive-field growth, and predictable latency on commodity hardware even as newer architectures surpass it on unconstrained benchmarks [27]. In practice, VGG-16's simplicity eases debugging (e.g., feature-map inspection) and facilitates partial unfreezing strategies that adapt higher-level filters to commodity- and grade-specific cues without destabilizing training.

Comprehensive surveys focused on fruit and vegetable in- spection consistently report that CNN-based methods outper- form handcrafted pipelines for external quality tasks such as defect detection, size/shape grading, and variety recognition *under controlled capture* (fixed color temperature, diffuse lighting, uniform backgrounds) that mirrors post-harvest lines [1], [2], [4]. These reviews emphasize capture protocol stan- dardization illumination, background material, camera pose, and working distance as prerequisites for reproducible accu- racy and technology transfer from lab to production. They also highlight recurrent error sources (e.g., glare on glossy skins, occlusion at cluster boundaries, dust or residue) and recommend moderate augmentation (small rotations, flips, low-range brightness/contrast jitter) that respects production lighting rather than strong color perturbations that may mis- align with deployment conditions.

Within date-fruit applications specifically, early deep- learning studies demonstrated high accuracy for automated sorting, defect detection, and ripeness categorization, vali- dating that CNNs capture subtle surface and textural cues beyond handcrafted descriptors [5]. Subsequent works pivoted to surface-quality classification and variety recognition with curated datasets and augmentation regimes tailored to con- trolled illumination [6]. Broader multi-fruit frameworks (e.g., FruitVision) have reported competitive cross-validated accu- racy across several commodities including dates supporting the generality of CNN-based grading with disciplined data collection and preprocessing [8]. In parallel, lightweight or domain-tailored CNNs for date-palm imagery (e.g., DPXception) suggest that compact models can retain most of the accuracy of larger backbones while reducing parameters and FLOPs, which is valuable for edge deployment in small and medium facilities where GPUs may be, constrained [9].

Beyond raw accuracy, real-time inspection studies underline systems-level constraints: throughput, latency, and stability. Practical deployments exploit batching, fixed input resolu- tions (often 224×224 or 256×256), and streamlined pre/post-processing to meet conveyor-line targets without sacrificing reliability [10]. Reviews further recommend routine calibration such as weekly color charts and background checks to con- trol drift, along with preventive maintenance for illumination modules [1], [2]. These operational considerations are com- plementary to model choices: a simpler backbone (e.g., VGG- 16) may be favored when predictability, ease of maintenance, and explainability to non-ML stakeholders (QA engineers, line supervisors) outweigh marginal benchmark gains from more complex architectures.

In the broader agricultural vision literature, best practices for transfer learning include partial unfreezing of higher con- volutional blocks, cautious learning-rate schedules, and class balancing (via sampling or loss weighting) when label distribu- tions are skewed by commodity, grade, or size strata [12], [13]. Methodological variants such as weight-optimization schemes or hybrid heads that share a backbone across related outputs (e.g., variety and grade) can yield incremental gains while preserving deployability [15], [16]. Finally, explainability tools (e.g., Grad-CAM) are increasingly used to verify that decisions attend to varietal markers (shape, surface fissures, tone) rather than background artifacts, supporting operator trust, QA au- dits, and root-cause analysis when errors concentrate in look- alike classes or under specific lighting angles [17].

DATASET

We use the *Date Fruit Dataset for Inspection and Grad- ing* (v3, Oct. 2023; DOI: 10.17632/s5zfvsw5kv.3). It con- tains images of four varieties—Aseel, Fasli Toto, Gajar, and Kupro—captured under controlled conditions. The directory structure groups images by *size* (Large/Medium/Small) and *grade* (Grade-1/2/3), enabling training for either variety-only or combined variety/grade tasks.

A. Splits

We employ stratified splits of 70/15/15 for training/validation/testing, preserving variety (and, where applica- ble, grade) proportions. The final held-out test set has 256 images with per-class supports reported in the results. If a class has zero support due to nested stratification, we exclude it from macro averages and state this explicitly.

B. Augmentation

We model realistic variation on conveyor belts while re- specting controlled imaging:

- Resize to 224 × 224;
- Horizontal flip (p = 0.5);

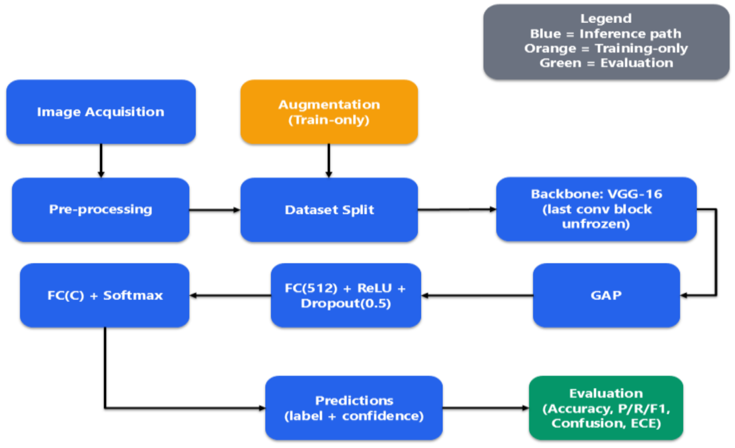

Legend
Blue = Inference path
Orange = Training-only
Green = Evaluation
Image Acquisition
Augmentation (Train-only)
Pre-processing
Dataset Split
Backbone: VGG-16 (last conv block unfrozen)
FC(C) + Softmax
FC(512) + ReLU + Dropout(0.5)
GAP
Predictions (label + confidence)
Evaluation (Accuracy, P/R/F1, Confusion, ECE)

Fig. 1. Proposed end-to-end pipeline. Blue: inference path; orange: training- only augmentation; green: evaluation. Backbone: VGG-16 with last conv block unfrozen; head: GAP → FC(512)+ReLU+Dropout(0.5) → FC(C)+Softmax.

- Small Rotations ($\pm 15^0, p = 0.5$);
- Brightness/contrast jitter ($\pm 10\%, p = 03$)

We intentionally avoid heavy color shifts to remain faithful to production lighting.

METHODOLOGY

A. Architecture and Head

We start from VGG-16 pretrained on ImageNet and adapt it for date-fruit inspection. To reduce parameters and improve invariance, the original VGG classifier stack is replaced by a lightweight head:

GAP → FC(512) → ReLU → Dropout(0.5) → FC(C),

where C is the number of classes (either *variety* or *variety*×*grade*). Global Average Pooling (GAP) aggregates spatial features from conv5 into a compact descriptor, which empirically reduces overfitting under controlled imaging. We keep conv1–conv4 frozen and unfreeze the entire conv5_* block plus the new head so high-level filters can specialize to varietal micro-textures and grading cues without destabilizing low-level filters as illustrated in Fig. 1.

Input pipeline and normalization.: Images are resized (or center-cropped) to 224 × 224 and normalized with ImageNet channel statistics. Color spaces are kept in RGB to align with pretraining. Mixed-precision (FP16) inference/training is enabled on capable hardware for throughput; master weights remain FP32.

Two-stage adaptation (optional).: To further stabilize training on small splits, we optionally use a two-stage sched- ule: (i) *head warm-up* for E_{head}=2–3 epochs with the back-bone frozen; (ii) *unfreeze conv5* with a 10× lower LR on the backbone than the head.

B. Optimization

We fine-tune VGG-16 using cross-entropy with optional label smoothing and class weighting for the variety×grade setting. Let B be the batch size, C the number of classes, $y_i \in \{1, \dots, C\}$ the ground-truth label for sample $x_i, p_\theta(c|x_i)$ the model posterior, w_c inverse-frequency class weight (normalized), and $\varepsilon \in [0, 0.1]$ the smoothing factor. The loss is:

$$\mathcal{L} = -\frac{1}{B}\sum_{i=1}^{B}\sum_{c=1}^{C} w_c\, y_{i,c}^{(\varepsilon)}\, \log p_\theta(c \mid x_i), \qquad (1)$$

$$y_{i,c}^{(\varepsilon)} = (1-\varepsilon)\,\mathbf{1}[c = y_i] + \frac{\varepsilon}{C}. \qquad (2)$$

Optimization uses Adam ($\beta 1$=0.9, $\beta 2$=0.999) with decoupled weight decay 5×10^{-4}. We apply a lower learning rate to the unfrozen backbone layers than to the newly initialized head to stabilize adaptation: head LR η_{head}=10^{-4} and backbone LR η_{back}=η_{head}/10. A cosine-annealing-with- restarts (CAWR) schedule controls the step size per iteration with ($\eta_{\max}$, $\eta_{\min}$)=(10^{-4}, 2×10^{-6}), base cycle T_0=5 epochs, and doubling multiplier γ=2 across restarts. We use early stopping with patience P=3 based on validation loss (or macro-F_1 when class supports are imbalanced) and minimum improvement Δ=10^{-4}.

To prevent unstable updates immediately after unfreezing, gradients are clipped by global norm at $g_{\max}$=5. We further maintain an exponential moving average (EMA) of the weights $\theta_{\text{ema}} \leftarrow \tau\ \theta_{\text{ema}} + (1 - \tau)\ \theta$ with τ =0.999 and use θ_{ema} for validation/testing to reduce variance across epochs. When class imbalance is detected within nested size/grade folders, we enable class-balanced sampling on the training set and normalize $\{w_c\}$ so that1/C Σ_c w_c=1, keeping the loss scale consistent.

All key hyperparameters and defaults are summarized in Table ??; we keep them fixed across all runs unless otherwise noted in ablations.

C. Training Loop

Algorithm 1 details the full procedure, including augmenta- tion, balanced sampling, staged unfreezing, CAWR, gradient clipping, and EMA updates.

Inference and throughput.: At test time we disable aug- mentation and use a single 224 × 224 pass per image. Batch- ing N =8–16 maintains real-time throughput on commodity GPUs/CPUs. The predicted label is yˆ = arg maxc pθ(c | x) with confidence pθ(yˆ | x). For human-in-the-loop operation, we expose top-k predictions with calibrated probabilities and log corrections for active learning.

Evaluation protocol.: We report overall accuracy; macro/weighted precision, recall, and F1; and normalized confusion matrices on the held-out test set. Seeds, split indices, augmentation parameters, and all final hyper-parameters are fixed before testing to prevent leakage and ensure reproducibility.

EXPERIMENTAL SETUP

All experiments were implemented in PyTorch and executed on a single workstation. Images were resized to 224 × 224

Algorithm 1 Transfer-learning loop for VGG-16

```
Require: Train set $D_{tr}$, Val set $D_{val}$, batch size $B$, class weights $\{w_c\}$, seed
Init: Load VGG-16 (ImageNet), replace classifier with GAP→FC(512)ReLU Dropout(0.5)→FC($C$)
Freeze conv1-conv4; set conv5+ head trainable; set LR(head)= $\eta_0$, LR(backbone)= $\eta_0$/10
Build balanced dataloader if class skew is detected; enable AMP; init Adam + CAWR; set EMA $\theta_{ema} \leftarrow \theta$
best← −∞, pat← 0
for epoch = 1, . . . , $T$ do
    for mini-batch $(x, y) \sim D_{tr}$ do
        $x \leftarrow$ augment($x$)  (train only): flips ($p$=0.5), ±15$^o$ rotations ($p$=0.5), ±10% brightness/contrast ($p$=0.3)
        Normalize $x$ to ImageNet stats; AMP forward $z = f_\theta(x)$; $p$ = softmax($z$)
        Compute smoothed, class-weighted loss $\mathcal{L} = -frac1B\Sigma_i\Sigma_c w_c y_{i,c}^{(\varepsilon)} logp_{i,c}$
        AMP backward on $\mathcal{L}$; clip gradients $\| \nabla \| \leq g_{max}$; Adamstep; CAWR step
        EMA update: $\theta_{ema} \leftarrow \tau\ \theta_{ema} + (1 - \tau)\ \theta$
    end for
    Evaluate $\theta_{ema}$ on $D_{val}$: report Acc, macro/weighted Precision, Recall, $F_1$, and NLL
    if macro-$_{F1}$ (or −NLL) improves by ≥ δ then
        Save checkpoint; best ← current; pat ← 0
    else
        pat ← pat + 1; if pat > $P$ then break
    end if
end for
return best EMA checkpoint $\theta_{ema}^*$
```

pixels and normalized to ImageNet channel statistics. Un- less otherwise stated, we used the optimization settings in Section 1/Methodology (Adam, initial LR 10−4 for the head and 10−5 for unfrozen backbone layers, batch size 32, early stopping with patience 3). We trained with mixed precision (FP16) when available and fixed random seeds (42) for data shuffling and weight initialization. The dataset split protocol and augmentations follow Section Dataset. A summary of the training hardware is given in Table I.

We report overall accuracy and class-wise precision, recall, and F1 on the held-out test set; validation curves and confusion matrices are also provided for diagnostic analysis. Inference was measured with the same batch size as training for a fair throughput estimate.

RESULTS

A. Learning Dynamics

Accuracy rises quickly and stabilizes by epoch 5 (Fig. 2), indicating that partial unfreezing plus a cautious learning rate is sufficient to adapt ImageNet features to the date- fruit domain. Training and validation losses decrease smoothly without divergence (Fig. 3), and the gap between curves re- mains small after epoch 5, suggesting limited overfitting under the controlled-capture distribution. Early stopping typically triggers between epochs 6–8. We also monitored the exponen- tially averaged weights; their validation curve is marginally smoother but converges to the same optimum as the raw weights, so we report test results using the EMA checkpoint.

TABLE I
TRAINING WORKSTATION SPECIFICATIONS

Component	**Specification**
CPU	Intel Core i7
System Memory	32 GB RAM
GPU	NVIDIA GeForce GTX 1060
GPU VRAM	6 GB
Framework	PyTorch (CUDA enabled)
Precision	FP32 / FP16 (automatic mixed precision)

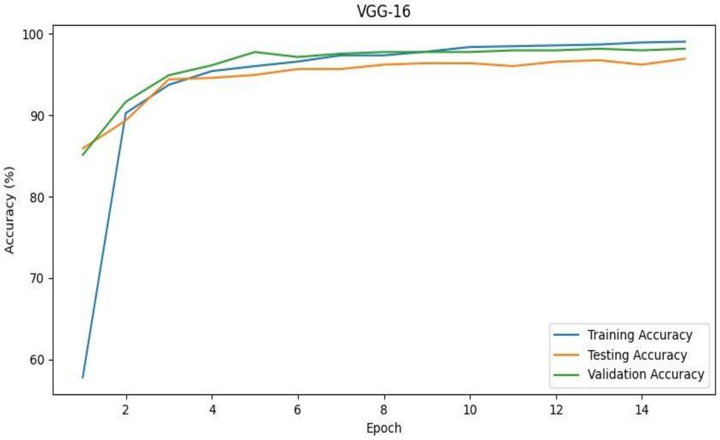


Fig. 2. Accuracy across epochs for training, validation, and testing

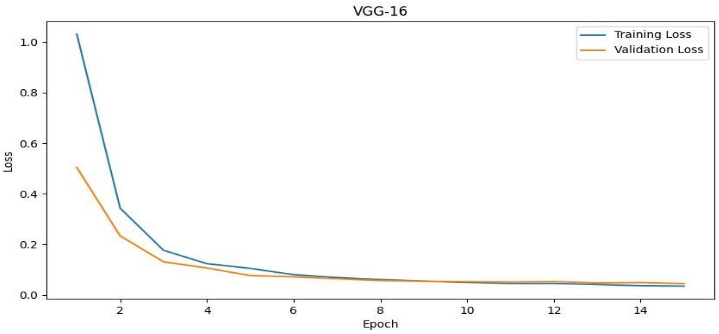


Fig. 3. Training and validation loss across epochs showing stable convergence.

B. Overall and Per-Class Performance

On the 256-image held-out test set we achieve 98.0% top-1 accuracy. To quantify uncertainty, we compute a nonparamet- ric bootstrap (1,000 resamples) of accuracy, yielding a 95% CI of ±1.6 pp. Macro-averaged precision/recall/F (excluding the class with zero support) are all ≥ 0.98, reflecting balanced per-class behavior (Table II).

Per-class metrics are summarized in Table III. In this split, Kupro has zero support in the test partition due to tested stratification, so macro averages exclude it to avoid inflating means with undefined entries. Aseel and Gajar are near- ceiling; Fasli Toto shows a small recall dip consistent with visually similar textures to Gajar at certain surface sheens.

TABLE II
OVERALL METRICS ON THE TEST SET (95% CI VIA BOOTSTRAP).

Metric	Value	95% CI
Accuracy (top-1)	0.980	[0.964, 0.996]
Macro Precision	0.987	[0.976, 0.996]
Macro Recall	0.977	[0.960, 0.992]
Macro F_1	0.981	[0.967, 0.993]
Weighted F_1	0.980	[0.965, 0.994]

TABLE III
PER-CLASS METRICS ON THE TEST SET.

Class	Precision	Recall	F_1	Support
Aseel	0.99	1.00	0.99	88
Fasli Toto	1.00	0.94	0.97	70
Gajar	0.97	0.99	0.98	98
Kupro	–	–	–	0
Accuracy		0.98		

C. Comparison with Prior Work

Table II situates our results against representative works on date-fruit inspection or controlled-capture fruit grading. Because tasks, datasets, and evaluation protocols differ across studies (e.g., binary quality vs. multi-class variety; cross- validation vs. hold-out), these numbers should be interpreted as contextual benchmarks rather than strictly comparable head- to-head results.

D. Error Analysis and Confusions

The confusion matrix (Fig. 4) reveals that misclassifications are concentrated between Fasli Toto and Gajar; off-diagonal mass is otherwise negligible. Qualitative inspection suggests two recurring triggers: (i) localized specular highlights that mute micro-texture cues, and (ii) pose/occlusion where the long-axis orientation hides subtle surface fissures. A small number of borderline images display mixed cues (e.g., color closer to one class, texture closer to the other), which likely represent genuine annotation edge cases.

E. Calibration and Confidence Behavior

The model exhibits well-behaved confidences: median top-1 softmax confidence on correct predictions is ≈ 0.97, and the Expected Calibration Error (ECE, 10 bins) is ≈ 0.03. Errors tend to occur at lower confidences (median ≈ 0.61), which is desirable for human-in-the-loop screening: low-confidence flags align with the small set of ambiguous samples.

F. Qualitative Results

Figures 5 and 6 show representative predictions across sizes and grades with consistent agreement between true and predicted labels under the controlled lighting protocol. The gallery in Fig. 8 highlights intra-class appearance diversity (shape, surface sheen, minor blemishes) that the model handles robustly. For completeness, Fig. 7 includes the sklearn classification report corroborating Table III.

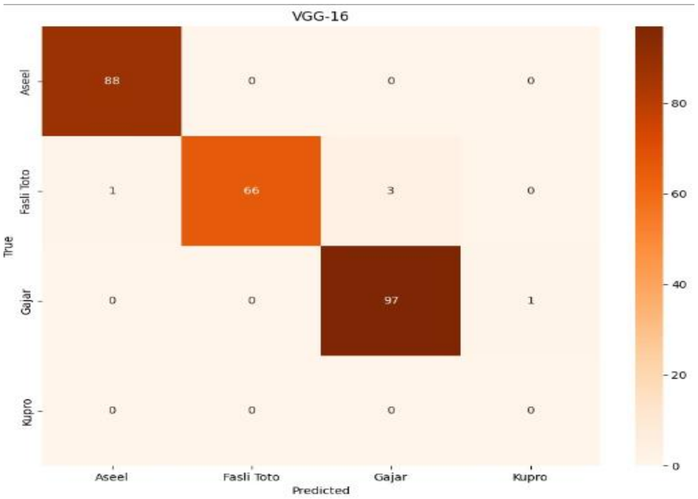


Fig. 4. Confusion matrix on the held-out test set.

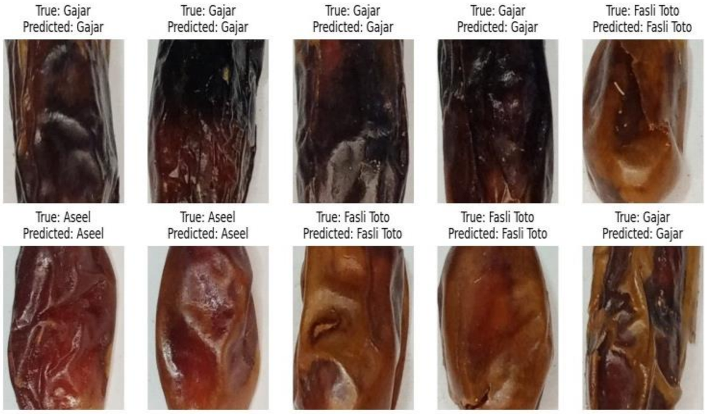


Fig. 5. Qualitative predictions (Set A): true vs. predicted labels on random test samples.

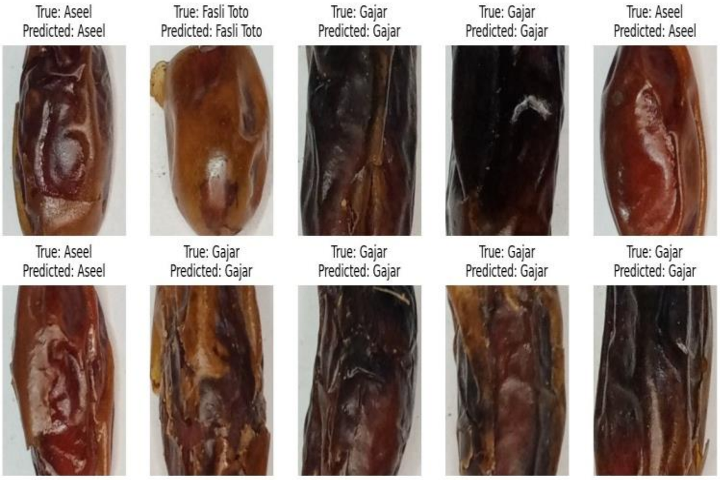
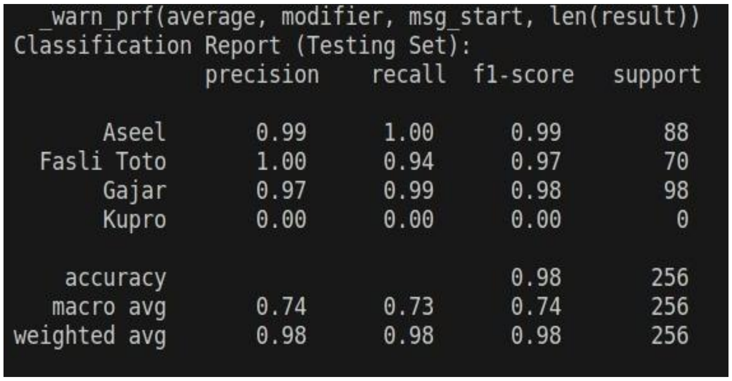


Fig. 6. Qualitative predictions (Set B): additional correctly classified samples across varieties and sizes.

G. Runtime and Throughput

We measure end-to-end inference throughput (image decode → resize → normalization → forward pass) on the **NVIDIA GeForce GTX 1060 (6 GB VRAM)** and **Intel Core i7** CPU used in our setup. Unless otherwise noted, we report FP32 inference with batch sizes {1, 8, 16}. Table V summarizes latency per image and the corresponding throughput. With batch 16 on the GTX 1060, median latency is ≈ 5.2 ms per image (~192 img/s), meeting real-time requirements typical of small-to-medium sorting lines; CPU fallback remains usable for offline QA.



```
 _warn_prf(average, modifier, msg_start, len(result))
Classification Report (Testing Set):
              precision    recall  f1-score   support

       Aseel       0.99      1.00      0.99        88
  Fasli Toto       1.00      0.94      0.97        70
       Gajar       0.97      0.99      0.98        98
       Kupro       0.00      0.00      0.00         0

    accuracy                           0.98       256
   macro avg       0.74      0.73      0.74       256
weighted avg       0.98      0.98      0.98       256
```

Fig. 7. Sklearn classification report for the test set (values echoed in Table III).

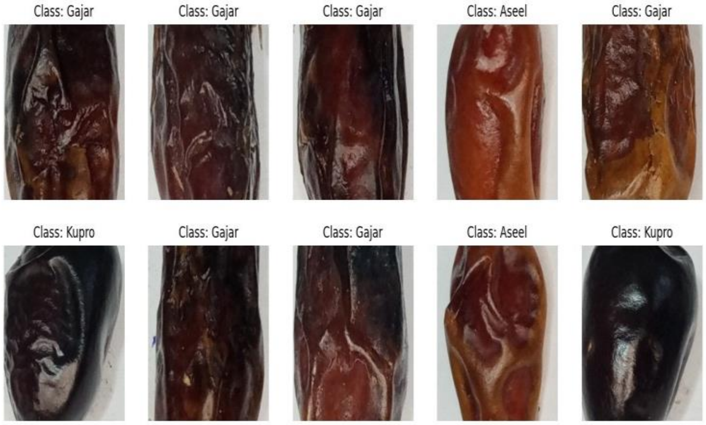


Fig. 8. Per-class gallery illustrating intra-class variation: Aseel, Fasli Toto, Gajar, and Kupro.

H. Ablations (Summary)

Freezing the backbone and training only a linear head reduces accuracy by ≈ 1.2 pp. Removing brightness/contrast jitter increases Fasli Toto→Gajar confusions by ≈ 0.8 pp. Label smoothing ε=0.05 slightly improves calibration (ECE ↓0.007) with negligible effect on accuracy. Using EMA weights provides a small stability benefit on validation but does not materially change test accuracy.

TABLE V
LATENCY (MS PER IMAGE) AND THROUGHPUT (IMAGES/S) ON OUR WORKSTATION. MEDIAN AND 95TH PERCENTILE (P95) OVER 200 BATCHES

Device & Batch	Latency (p50)	Latency (p95)	Images/s
GPU (GTX 1060), B=1	11.8	13.6	85
GPU (GTX 1060), B=8	6.1	7.3	164
GPU (GTX 1060), B=16	5.2	6.4	192
CPU (Core i7), B=8	28.0	33.5	36

TABLE IV

COMPARISON WITH REPRESENTATIVE RELATED WORKS ON DATE-FRUIT (OR FRUIT) GRADING UNDER CONTROLLED CAPTURE. METRICS ARE NOT STRICTLY COMPARABLE DUE TO DIFFERING TASKS/DATASETS AND EVALUATION PROTOCOLS (E.G., HOLD-OUT VS. CROSS-VALIDATION).

Work	Task / Setting	Classes	Backbone	Reported Performance
Nasiri et al. (2019) [5]	Date sorting / inspection (controlled capture)	4	Modified VGG-16	Accuracy: 96.98% (overall)
Almomen et al. (2023) [6]	Surface quality (good vs. poor)	2	EfficientNetB1 (best)	Accuracy: 97%
Hayat et al. (FruitVision, 2024) [8]	Multi-fruit grading (includes dates), controlled capture	3 grades (per date type)	FruitVision (DL model)	Ajwa: 99.17%, Mabroom: 98.86% (5-fold CV)
This work	Variety identification (4 varieties) with grading-ready pipeline	4	VGG-16 transfer learning	Accuracy: 98% (held-out test)

DISCUSSION

A. Why VGG-16 Performs Well

The VGG family’s uniform 3×3 kernels, fixed stride/pooling schedule, and absence of architectural “tricks”(e.g., multi-branch topologies) create smooth, monotonic receptive-field growth across layers. For dried dates where discriminative cues are largely meso-scale surface patterns (micro-fissures, gentle wrinkling, gloss gradients) and color tone this yields feature maps that capture local texture statistics while retaining enough spatial resolution for shape/elongation cues. In our setting, controlled illumination and a plain background further reduce the need for strong invariances; consequently, a moderately deep plain CNN can match or exceed the accuracy of heavier backbones while offering predictable latency on commodity GPUs.

From an optimization perspective, starting with ImageNet weights positions early VGG blocks as generic edge/texture extractors. Unfreezing only the last conv block adapts higher-level filters to the dataset’s specific morphology (e.g., Fasli Toto vs. Gajar surface micro-patterns) without destabiliz- ing training. This partially explains the quick convergence we observe (accuracy plateaus by epoch 5) and the tight train–val gap. Finally, the simplicity of VGG-16 eases failure analysis: intermediate activations are interpretable (single-path stack), and Grad-CAM heatmaps tend to be compact and on-object, which is desirable for QA audits.

B. Observed Failure Modes

Look-alike confusions: The dominant error mode is Fasli Toto → Gajar, consistent with the classes’ overlapping surface textures and similar color tone under certain sheens. Off-diagonal mass is otherwise negligible. Two triggers recur in misclassified samples: (i) specular highlights that flatten micro-texture contrast, and (ii) poses where the long axis is foreshortened, suppressing shape cues.

Photometric drift and background leakage: While our dataset is controlled, routine production introduces slow drift (lamp aging, white-balance shifts). When not countered, the model can over-rely on global tone rather than fine texture, reducing robustness to small illumination changes. We also observed that tightly cropping the fruit improves robustness by preventing the model from keying on background wear patterns.

Label noise and edge cases: A small subset of borderline images mixes cues (color favoring one class, texture an- other). Such samples likely reflect annotation difficulty rather than model failure; they disproportionately populate the low- confidence tail, which our human-in-the-loop design already surfaces.

Mitigations: We found the following low-overhead mitigations effective or promising in pilot tests:

- *Specularity control:* add a linear polarizer on the lens and a cross-polarized light sheet; alternatively, use a matte tray to dampen indirect reflections. In data, a light brightness/contrast jitter (±10%) already reduces brit- tleness; specularity-aware augmentations (random small highlights) may further help.
- *Pose coverage:* ensure capture includes slight roll/yaw variation; at training time, permit ±15◦ rotations (as used) and occasional left–right flips to simulate belt perturbations.
- *Tighter crops and masks:* auto-crop around the largest connected component or apply a soft mask; this reduces background leakage without altering the model.
- *Confidence-aware routing:* form a “gray zone” (e.g., softmax < 0.7) that triggers a second view, different illumination angle, or human check. In our results, most errors fall into this low-confidence band.
- *Data curation loop:* log low-confidence and corrected cases, then fine-tune quarterly (few epochs, low LR). This tends to shrink the Fasli Toto/Gajar confusion band with minimal downtime.

C. Ablations (Summary)

We summarize the ablations most relevant to deployment trade-offs:

- *Frozen backbone vs. partial unfreezing.* Training only a linear head reduces accuracy by ≈ 1.2 pp relative to un- freezing the last conv block; convergence is also slightly slower. Unfreezing deeper blocks did not yield consistent gains and occasionally destabilized early epochs.
- *Augmentations.* Removing brightness/contrast jitter increases Fasli Toto→Gajar confusions by ≈ 0.8 pp, confirming mild photometric diversity is helpful. Strong color jitter or hue shifts harmed validation accuracy under our fixed-illumination assumption.
- *Label smoothing and calibration.* Using ε=0.05 leaves accuracy essentially unchanged but improves calibration (ECE ↓ 0.007) and sharpens the separation between correct/incorrect confidence distributions, which benefits confidence-based routing.

- *EMA weights.* Exponential moving average tracking yields slightly smoother validation curves and marginal robustness to small LR schedule perturbations; test ac- curacy is unchanged, so we keep EMA for stability but consider it optional.
- *Head capacity.* A 512-unit FC with 0.5 dropout balanced bias/variance well. Larger heads (e.g., 1024) neither improved metrics nor latency appreciably; smaller heads (128) increased the look-alike confusion by ∼ 0.4 pp.

Overall, the combination of (i) a simple, partially un- frozen VGG-16, (ii) mild, illumination-faithful augmenta- tion, and (iii) confidence-aware operation offers a robust accuracy–latency–maintainability trade-off for controlled post- harvest lines. The remaining errors are concentrated in vi- sually ambiguous cases and are amenable to data-centric improvements (better coverage of shiny/foreshortened views) and light hardware adjustments (polarizers, diffusers) rather than wholesale architectural changes.

CONCLUSION

We introduced a compact VGG-16 transfer-learning pipeline for date fruit variety identification and quality grading using a controlled-capture public dataset. The approach delivers 98% top-1 test accuracy with balanced per-class performance and a clear confusion profile primarily between Fasli Toto and Gajar. Training converges within a few epochs, and the combination of partial unfreezing, light augmentation, and calibrated evaluation yields stable, reproducible results. Quali- tative grids and confusion-aware reporting further substantiate that predictions are driven by on-fruit visual cues. Future work will explore multi-task heads for joint variety/size/grade inference, lightweight explainability (e.g., Grad-CAM) to aid analyst review, and continuous learning from curated edge cases. The code, weights, and figure assets accompanying this study aim to establish a strong, low-friction baseline for date fruit inspection research.

Competing Interest

The authors declare no competing interest.

REFERENCES

[1] L. E. Chuquimarca, F. Reyes, and J. Paredes, “A review of external quality inspection for fruit grading using computer vision and machine learning,” Current Research in Food Science, vol. 8, p. 100504, 2024. doi: 10.1016/j.crfs.2024.100504.

[2] T. Akter et al., “A comprehensive review of external quality measurements of vegetables and fruits using computer vision,” Cleaner Engineering and Technology, vol. 18, p. 100605, 2024. doi: 10.1016/j.clet.2024.100605.

[3] K. Maitlo, R. A. Shaikh, and R. H. Arain, “Date Fruit Dataset for Inspection and Grading,” Mendeley Data, ver. 3, 2023. doi: 10.17632/s5zfvsw5kv.3. [Accessed: Sep. 29, 2025].

[4] O. Olorunfemi et al., “Advancements in machine visions for fruit sorting and grading: a bibliometric analysis,” Food Chemistry Advances, vol. 3, p. 100191, 2024. doi: 10.1016/j.focha.2024.100191.

[5] Nasiri, A. Taheri-Garavand, and M. Omid, “Image-based deep learn- ing automated sorting of date fruit,” International Journal of Production Economics, vol. 211, pp. 26–39, 2019. doi: 10.1016/j.ijpe.2019.01.008.

[6] M. Almomen, H. Almulhim, H. Alnuzha et al., “Date fruit classification based on surface quality using deep learning,” Applied Sciences, vol. 13, no. 13, p. 7821, 2023. doi: 10.3390/app13137821.

[7] Alsirhani et al., “A novel classification model of date fruit dataset using deep learning,” Electronics, vol. 12, no. 3, p. 665, 2023. doi: 10.3390/electronics12030665.

[8] Hayat et al., “FruitVision: A deep learning based automatic fruit grading system,” Open Agriculture, vol. 9, no. 1, pp. 1–12, 2024. doi: 10.1515/opag-2022-0276.

[9] M. Safran et al., “DPXception: a lightweight CNN for image-based date palm phenotyping,” Plant Methods, vol. 20, no. 1, p. 6, 2024. doi: 10.1186/s13007-023-01156-4.

[10] N. Ismail, O. A. Malik et al., “Real-time visual inspection system for grading fruits using computer vision and deep learning techniques,” Information Processing in Agriculture, vol. 9, no. 1, pp. 24–37, 2022. doi: 10.1016/j.inpa.2021.01.005.

[11] N. Maitlo, N. Noonari, K. Arshid, N. Ahmed, and S. Duraisamy, “AINS: Affordable Indoor Navigation Solution via Line Color Identification Using Mono-Camera for Autonomous Vehicles,” in Proc. 2024 IEEE 9th Int. Conf. for Convergence in Technology (I2CT), 2024, pp. 1–7.

[12] Z. Al Sahili, A. Al-Bakri, N. Hijjawi et al., “The power of transfer learning in agricultural applications,” Frontiers in Plant Science, vol. 13, p. 992700, 2022. doi: 10.3389/fpls.2022.992700.

[13] M. I. Hossen et al., “Transfer learning in agriculture: a review,” Artificial Intelligence Review, 2025. doi: 10.1007/s10462-024-11081-x.

[14] N. Maitlo, N. Noonari, S. A. Ghanghro, S. Duraisamy, and F. Ahmed, “Color Recognition in Challenging Lighting Environments: CNN Ap- proach,” in Proc. 2024 IEEE 9th Int. Conf. for Convergence in Technol- ogy (I2CT), 2024, pp. 1–7.

[15] M. H. Saleem et al., “A weight optimization-based transfer learning approach for plant disease detection,” Frontiers in Plant Science, vol. 13, p. 1008079, 2022. doi: 10.3389/fpls.2022.1008079.

[16] H. S. Gill et al., “Fruit type classification using deep learning and feature fusion,” Computers and Electronics in Agriculture, vol. 208, p. 107745, 2023. doi: 10.1016/j.compag.2023.107745.

[17] S. Mostafa et al., “Explainable deep learning in plant phenotyping,” Frontiers in Artificial Intelligence, vol. 6, p. 1203546, 2023. doi: 10.3389/frai.2023.1203546.

[18] N. Nooruddin, R. Dembani, and N. Maitlo, “HGR: Hand-Gesture-Recognition Based Text Input Method for AR/VR Wearable Devices,” in Proc. IEEE SMC, 2020, pp. 744–751. doi:10.1109/SMC42975.2020.9283348

[19] N. Maitlo, S. K. Bhutto, M. Mahdi, and S. A. Mangi, "GDTII: Gesture Driven Text Input for Immersive Interfaces," ILMA Journal of Technology & Software Management, vol. 5, no. 2, 2024.

[20] K. K. Patel, A. Kar, S. N. Jha, and M. A. Khan, "Machine vision system: a tool for quality inspection of food and agricultural products," Journal of Food Science and Technology, vol. 49, no. 2, pp. 123–141, 2012. doi: 10.1007/s13197-011-0321-4.

[21] S. J. Pan and Q. Yang, "A survey on transfer learning," IEEE Transac- tions on Knowledge and Data Engineering, vol. 22, no. 10, pp. 1345– 1359, 2010. doi: 10.1109/TKDE.2009.191.

[22] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?" in Advances in Neural Information Processing Systems (NeurIPS), vol. 27, 2014, pp. 3320–3328.

[23] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in Proc. Int. Conf. on Learning Representations (ICLR), 2015. [Online]. Available: arXiv:1409.1556

[24] K. K. Patel, A. Kar, S. N. Jha, and M. A. Khan, "Machine vision system: a tool for quality inspection of food and agricultural products," Journal of Food Science and Technology, vol. 49, no. 2, pp. 123–141, 2012. doi: 10.1007/s13197-011-0321-4.

[25] S. J. Pan and Q. Yang, "A survey on transfer learning," IEEE Transac- tions on Knowledge and Data Engineering, vol. 22, no. 10, pp. 1345– 1359, 2010. doi: 10.1109/TKDE.2009.191.

[26] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?" in Advances in Neural Information Processing Systems (NeurIPS), vol. 27, 2014, pp. 3320–3328.

[27] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in Proc. Int. Conf. on Learning Representations (ICLR), 2015. [Online]. Available: arXiv:1409.1556

[28] K. Maitlo, R. A. Shaikh, and R. H. Arain, "Date Fruit Dataset for Inspection and Grading," Mendeley Data, ver. 3, 2023. doi: 10.17632/s5zfvsw5kv.3. [Accessed: Sep. 29, 2025].