Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Specimen-level validation of smartphone-based freshness classification of cow meat

Peter Appiahene
University of Energy and Natural Resources
Samuel Opoku Berchie
soberchie@gmail.com
University of Energy and Natural Resources
Henry Nii-Armah Mettle
University of Energy and Natural Resources
Rose-Mary Owusuaa Mensah Gyening
Kwame Nkrumah University of Science and Technology
Emmanuel Botchway
University of Energy and Natural Resources
James Quayson
University of Energy and Natural Resources
Philip Sarfo-Manu
University of Energy and Natural Resources
James Dorwodoe
University of Energy and Natural Resources
Benedicta Nyarko Anin-Acquah
University of Energy and Natural Resources
Priscilla Akoto Peprah
University of Energy and Natural Resources
Emmanuel Kwame Amoako
University of Energy and Natural Resources

Research Article

Keywords: Meat freshness, Image analysis, Data leakage, Explainable artificial intelligence, Model compression, Smartphone

Posted Date: November 25th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-11763888/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

Specimen-level validation of smartphone-based freshness classification of cow meat

Peter Appiahene[a], Samuel Opoku Berchie[a*], Henry Nii-Armah Mettle[a], Rose-Mary Owusuaa Mensah Gyening[b], Emmanuel Botchway[a], James Quayson[a], Philip Sarfo-Manu[a], James Dorwodoe[a], Benedicta Nyarko Anin-Acquah[a], Priscilla Akoto Peprah[a] and Emmanuel Kwame Amoako[a]

[a] Department of Information Technology and Decision Sciences, University of Energy and Natural Resources, Sunyani, Ghana

[b] Department of Computer Science, Kwame Nkrumah University of Science and Technology, Kumasi, Ghana

*** Corresponding author:** Samuel Opoku Berchie, Department of Information Technology and Decision Sciences, University of Energy and Natural Resources, Sunyani, Ghana; e-mail: soberchie@gmail.com; ORCID: https://orcid.org/0009-0006-2910-1149.

Abstract

Image-based deep learning has reported 93–100% accuracy for meat freshness, but freshness datasets contain many images of the same specimen and are usually split at random by image. We audited MeatScan, 10,518 smartphone images of fresh and spoiled cow meat from Ghanaian markets, and found that a split by capture session, which removes all near-duplicates detectable by perceptual hashing, still leaves a training image of the same piece of meat for 74% of test images. We built a specimen-level split by linking sessions with instance-level retrieval and geometric verification, and cropped every image to the meat region. On this split, the test macro-F1 of ImageNet-pretrained lightweight networks fell from 0.994–0.996 to 0.901–0.929. MeatGuard-S, a 1.14-million-parameter convolution–transformer hybrid trained without pretraining, reached 0.848–0.862. Distillation did not improve it; after pruning and 8-bit quantisation, it ran offline on an Android device in a median of 31.8 ms per image from a 0.95 MB file, but remained less accurate than the pretrained networks. Gradient-weighted class activation mapping (Grad-CAM), SHapley Additive exPlanations (SHAP) and occlusion maps, evaluated with deletion and insertion curves and the share of attribution on the meat, and an error analysis revealed a residual cue: the extent of the meat region carries label information after cropping, and all models rely on it on unseen specimens. Perturbation metrics that fill removed pixels with the mean colour could not detect this cue, and their faithfulness verdicts reversed between classes. Freshness datasets should be split by specimen, and explanations evaluated quantitatively.

Keywords: Meat freshness; Image analysis; Data leakage; Explainable artificial intelligence; Model compression; Smartphone

1. Introduction

Meat is highly perishable: after slaughter, microbial growth, lipid oxidation and enzymatic autolysis progressively change its pH, odour, surface texture and appearance (Dave & Ghaly, 2011). The burden of foodborne disease is highest in the African subregions (Havelaar et al.,

2015), and in low- and middle-income countries much of it is attributable to fresh foods sold through informal markets that escape effective regulation (Grace, 2015). Inspection in African slaughterhouses is often inadequate (Adesola et al., 2024), and freshness at the point of sale is judged by the senses, which is subjective (Kang et al., 2025). Sensory and chemical assays are time-consuming and destructive (Han et al., 2026), and spectroscopic and colorimetric methods demand specialised skills and lengthy procedures (Kang et al., 2025). Smartphones, which combine a capable camera, on-board computing and a familiar interface, are an attractive platform for on-site assessment (Kalinowska et al., 2021).

Image-based deep learning has accordingly reported accuracies of 93–100% for beef, chicken and fish freshness from red–green–blue (RGB) images (Abd Elfattah et al., 2025; Bhuiyan et al., 2024; Bramantyo et al., 2026; Hidalgo et al., 2025; Jian et al., 2025; Taheri-Garavand et al., 2020), and recent studies have moved towards compact networks and deployment outside the laboratory (Cao-Phan et al., 2026; Kozan & Akyürek, 2024; Song et al., 2024; Xu et al., 2026; Zhou et al., 2025).

Three limitations constrain how far these results can be trusted and used. First, freshness datasets contain many images of the same specimen, taken on successive days, from several angles or in quick succession, yet they are usually split at random by image (e.g. Hamidy et al., 2025; Song et al., 2024). When views of one specimen fall on both sides of the split, test accuracy partly measures the recognition of specimens already seen; only a few studies hold out whole specimens (Deliçay et al., 2026; Taheri-Garavand et al., 2020) or remove near-duplicate frames (Bramantyo et al., 2026). Backgrounds such as trays and packaging can likewise drive the predictions (Golder et al., 2024; Külcü & Balpetek Külcü, 2026). Second, compactness is usually reported as parameter counts or desktop throughput rather than measured on the target device

(Section 2.3), and distillation combined with pruning and 8-bit quantisation has not, to our knowledge, been examined for meat freshness. Third, explanations are typically shown as example heat maps and are seldom evaluated formally (Arrighi et al., 2026).

MeatScan (Gyening et al., 2025), about 11,000 smartphone images of fresh and spoiled cow meat from open-air markets, butcher shops and cold rooms in Ghana, is valuable because it was collected in real trading conditions, without controlled lighting or backgrounds. Auditing it before training, we found both of the problems above. Under a random 70/15/15 split at image level, 68% of test images have a perceptual near-duplicate in the training set. Grouping the frames into capture sessions removes these near-duplicates but not the repeated specimens: with whole sessions assigned to partitions, 74% of test images still have a training image of the same piece of meat, photographed again at another distance or angle, or mirrored, and 95% belong to a piece of meat with other images in the training set. In addition, a single capture session, mostly one piece of meat in a black polythene bag, accounts for about 54% of the spoiled images, so the background carries information about the label. Results obtained on MeatScan with random image-level splits, including 100% test accuracy (Faridi, 2026) and our own earlier benchmark (Mettle et al., 2026), are therefore likely to overstate performance on unseen pieces of meat.

This paper re-examines image-based freshness classification of cow meat under an evaluation protocol that removes both sources of optimism, and asks three questions. How accurate are compact models once leakage and background cues are controlled? Can distillation, pruning and quantisation make a purpose-built network without ImageNet pretraining competitive with pretrained lightweight networks? And what do quantitatively evaluated explanations reveal about the cues the models use? The contributions are as follows:

- **An evaluation framework that applies mechanisms to control data leakage:** A specimen-level split of MeatScan, obtained by linking capture sessions that show the same piece of meat with instance-level retrieval and geometric verification, combined with segmentation-based removal of the background.
- **A controlled comparison of compact models:** Three ImageNet-pretrained lightweight networks reach a test macro-F1 of 0.901 to 0.929, with confidence intervals and paired tests that resample specimens rather than images. MeatGuard-S, a convolution–transformer hybrid with 1.14 million parameters trained without pretraining, reaches 0.848 to 0.862. Distillation does not improve it. It keeps its accuracy when pruned to 0.85 million parameters and quantised to 8 bits, and it remains less accurate than the pretrained networks.
- **Measured explanations:** Grad-CAM, SHAP and occlusion maps evaluated with deletion and insertion curves, the share of attribution on the meat and a weight-randomisation check. it.
- **A simple mobile application which deploys the model.**

The rest of the paper is organised as follows: Section 2 reviews related work, Section 3 describes the methods, Section 4 reports the results, Section 5 discusses them together with the limitations of the study, and Section 6 concludes.

2. Related work

2.1 Non-destructive assessment of meat freshness

Meat spoils through microbial growth, lipid oxidation and enzymatic autolysis (Dave & Ghaly, 2011), and consumers judge the result by discoloration, off-odours and slime, a subjective judgement (Nychas et al., 2008). Laboratory indicators such as total volatile basic nitrogen (TVB-N) and total viable counts are objective but time-consuming and destructive (Han et al., 2026). Non-destructive alternatives include hyperspectral imaging (Sherin et al., 2026), spectroscopy and electronic noses (Kang et al., 2025), indicator films read with a smartphone

(Savarolyia et al., 2026) and the fusion of gas sensors with images (Bhuiyan et al., 2024). They track the chemistry of spoilage closely but need specialised skills or hardware, and laboratory accuracies have proved difficult to reproduce in real-world settings (Kang et al., 2025; Savarolyia et al., 2026). Computer vision with ordinary cameras is objective, fast and inexpensive (Modzelewska-Kapituła & Jun, 2022), camera-derived colour tracks storage time in beef more sensitively than a colorimeter (Chen et al., 2026), and visual appearance is the cue on which buyers and vendors already rely, which makes RGB imaging the most deployable option for informal markets.

2.2 RGB image-based deep learning for meat and fish freshness

Early approaches classified hand-crafted colour and texture features with conventional learners, reaching 93.01% accuracy for common carp (Taheri-Garavand et al., 2019) and 92.59% for beef (Arsalane et al., 2024). Convolutional networks have since replaced hand-crafted features, mostly through transfer learning from ImageNet-pretrained backbones (Abd Elfattah et al., 2025; Deliçay et al., 2026; Hamidy et al., 2025; Hidalgo et al., 2025; Jian et al., 2025; Lin et al., 2025; Taheri-Garavand et al., 2020), alongside custom networks (Bhuiyan et al., 2024), few-shot learning (C. Liu et al., 2025) and lightweight architectures (Section 2.3). Table 1 summarises representative studies; reported accuracies mostly lie between 93% and 100%.

Table 1

Representative RGB image-based studies of meat and fish freshness

Study	Product; data	Model	Split unit (protocol)	Background handling	Reported result	Deployment
Taheri-Garavand et al. (2020)	Carp; 672 images of 48 fish, illumination chamber	VGG-16 features + dense classifier	Specimen (4 whole fish held out)	Controlled chamber	98.21% accuracy	–
Bhuiyan et al. (2024)	Beef, mutton; 6,672 freshness images + gas sensors	Custom CNN	Image (10-fold CV)	Not addressed	99% accuracy	IoT unit with display
Song et al. (2024)	Pork; 2,757 images	MobileNetV3-Small (best of 5 CNN families)	Image (random 8:1:1)	Uniform white background	98.59% accuracy	Desktop client
Kozan and Akyürek (2024)	Meat; 1,896-image public set, augmented	Pretrained CNN	Not reported	Not addressed	99.39% validation accuracy	Mobile application
Hidalgo et al. (2025)	Beef, chicken; three public sets	Pretrained CNN features + RADAM + SVM/RF/MLP	Image (5- and 20-fold CV)	Close-range images, resizing	93–100%	–
Abd Elfattah et al. (2025)	Meat, three classes	VGG19 features + IAPO feature selection	Not reported	Not reported	98.51% accuracy	–

Study	Product; data	Model	Split unit (protocol)	Background handling	Reported result	Deployment
Zhou et al. (2025)	Meat; 2,266-image public set + lighting variation	MobileNetV3 + ECA	Image (5-fold CV)	Not addressed	98.6% accuracy	17.34 MB model
Yu et al. (2025)	Beef; 1,523 images, green screen, 10 days	DEVA-ConvNeXt	Image (8:1:1 per class)	Synthetic background replacement	94.8% accuracy	Embedded test
Jian et al. (2025)	Chicken breast; 3,820 images, 7 levels	YOLOv8n_CA_DSC3	Image (8:2)	Not addressed	95.6% accuracy	–
C. Liu et al. (2025)	Pork; 600 images graded by microbial count	BBSNet (few-shot)	Image (stratified 5-fold CV)	Not addressed	96.36% (5-way 80-shot)	–
Hamidy et al. (2025)	Sea bass; 10 fish, 8 views, daily	Seven transfer-learning CNNs	Image (random 80/20)	With and without background compared	98.94% (purchasability)	–
Bramantyo et al. (2026)	Meat; 6,256 images after deduplication	Five CNN and transformer backbones	Image, after pHash near-duplicate removal	U-Net segmentation	98.10% accuracy	Smartphone (TFLite)

Study	Product; data	Model	Split unit (protocol)	Background handling	Reported result	Deployment
Cao-Phan et al. (2026)	Pork; 10,372 images, merged public sets	DW-SPPFNet (1.63 M parameters)	Image (80/20 + 3-fold CV)	Not reported	98.31% accuracy	2.72 ms/image, device not stated
Deliçay et al. (2026)	Sea bass fillets; 430 images of 10 fish	DenseNet121 and four other CNNs	Specimen (leave-one-fish-out)	Controlled illumination	92.60% (purchasability, best)	–
Faridi (2026)	Beef; MeatScan, 10,517 smartphone images	ConvNeXt-Base + test-time augmentation	Image (stratified 75/10/15)	Not addressed	100% accuracy	–
Külcü and Balpetek Külcü (2026)	Beef; 2,266-image public set	MobileNetV2 + attention, Xception features, SVM/XGBoost ensemble	Image (fixed split + 5-fold CV)	Grad-CAM-guided suppression of trays	99.78% accuracy	GPU throughput
Xu et al. (2026)	Beef; 420 pieces from 20 cattle	RT-BFE (1.11 M parameters)	Not reported in available text (5-fold CV)	Laboratory capture	95.24% accuracy	Smartphone 32 ms; embedded board 14 ms (FP16)

Note. CV = cross-validation; "Image" = images assigned to partitions individually, without grouping by specimen or capture session; "Specimen" = all images of a specimen kept in one partition. Results are as reported by the authors and are not comparable across rows, because tasks, classes and datasets differ.

Most of these datasets were collected under controlled conditions, against a white background (Song et al., 2024), a green screen (Yu et al., 2025), in an illumination chamber (Taheri-Garavand

et al., 2020) or in the laboratory (Xu et al., 2026), and many studies reuse two small public collections, one of them taken by a fixed camera photographing a single tray of meat every two minutes (Ulucan et al., 2019). MeatScan departs from both patterns (Gyening et al., 2025); the only study we found that trains on it reports 100% test accuracy with a stratified random image-level split (Faridi, 2026).

The column that most affects the interpretation of Table 1 is the split unit. Most studies assign individual images to partitions at random although their datasets contain many images of each specimen, so near-identical views of a specimen appear on both sides of the split. Taheri-Garavand et al. (2020) and Deliçay et al. (2026) held out whole fish, but under laboratory conditions and with few specimens (48 and 10 fish); Bramantyo et al. (2026) removed near-duplicate frames by perceptual hashing before a random split, which reduces but does not remove the overlap between related images.

Backgrounds are the second source of optimism. Grad-CAM has shown models attending to packaging trays (Külcü & Balpetek Külcü, 2026) or to the wrong regions of fish images (Golder et al., 2024), and edge- and background-associated activations persist even under specimen-level evaluation (Deliçay et al., 2026). Proposed remedies include segmenting the meat (Bramantyo et al., 2026), flesh-only patches (Lin et al., 2025), synthetic background replacement (Yu et al., 2025) and attention steering with explanation maps (Külcü & Balpetek Külcü, 2026). None of the studies we reviewed combines a split by specimen or capture session with background removal on images collected in real trading conditions.

2.3 Lightweight architectures and model compression

Mobile vision models trade accuracy against computation and latency. MobileNets factorise convolutions into depthwise and pointwise operations (Howard et al., 2017; Sandler et al., 2018),

and MobileNetV3 adds architecture search and squeeze-and-excitation attention (Howard et al., 2019; Hu et al., 2018); ShuffleNet V2 argued that speed should be measured on the target platform rather than inferred from operation counts (Ma et al., 2018); and EfficientNet scales depth, width and resolution jointly (Tan & Le, 2019). Hybrid convolution–transformer networks bring the global context of vision transformers (Dosovitskiy et al., 2021; Z. Liu et al., 2021) to the mobile regime (Li et al., 2022; Maaz et al., 2023; Mehta & Rastegari, 2022).

Compression offers a complementary route. Knowledge distillation trains a small student on the softened outputs (Hinton et al., 2015) and intermediate representations (Romero et al., 2015) of a larger teacher (see Gou et al., 2021, for a survey); filter pruning removes convolutional filters with small $\ell 1$-norms (Li et al., 2017); and pruning and quantisation together reduce storage many-fold without loss of accuracy (Han et al., 2016) and allow integer-only inference (Jacob et al., 2018).

Freshness studies use lightweight backbones directly or with added attention (Song et al., 2024; Zhou et al., 2025), and purpose-built networks reach 1.1 to 1.6 million parameters (Cao-Phan et al., 2026; Xu et al., 2026); Xu et al. (2026) measured 32 ms per image on a smartphone with a 16-bit model. Distillation has transferred knowledge from multispectral to RGB images for fish freshness (Tsagkatakis et al., 2021), but we found no meat-freshness study that combines distillation with structured pruning and 8-bit quantisation, and on-device latency remains the exception (Table 1).

2.4 Explainability for food-quality models

Grad-CAM weights the final convolutional feature maps by the gradients of the class score (Selvaraju et al., 2017), occlusion measures the change in output as regions are masked (Zeiler & Fergus, 2014), and LIME (Ribeiro et al., 2016) and SHAP (Lundberg & Lee, 2017) attribute

predictions to interpretable units such as SLIC superpixels (Achanta et al., 2012). Class-activation methods are the most common explanations in food applications (Arrighi et al., 2026), and freshness studies use Grad-CAM to show the regions behind a prediction (Deliçay et al., 2026; Xu et al., 2026) or to diagnose reliance on irrelevant regions (Golder et al., 2024; Külcü & Balpetek Külcü, 2026).

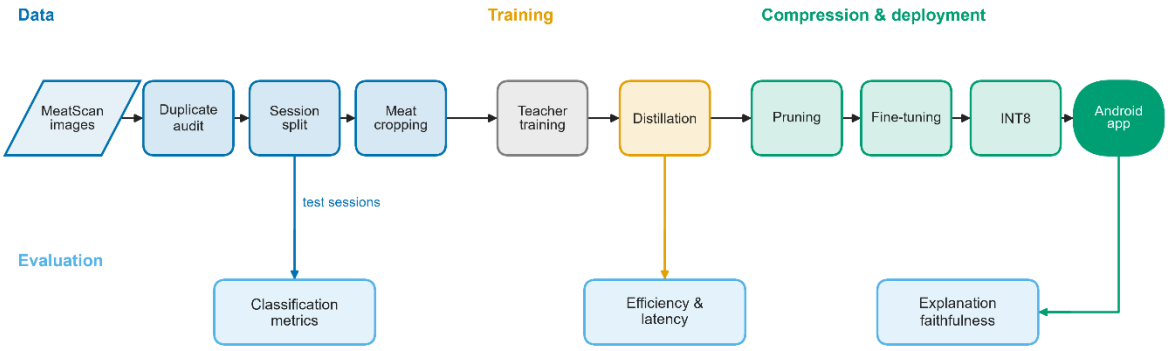
A plausible heat map is not evidence that it reflects the model's reasoning: some saliency methods produce maps independent of both the model parameters and the training data (Adebayo et al., 2018). Deletion and insertion metrics measure faithfulness by tracking the class probability as the most important pixels are removed or restored (Petsiuk et al., 2018), but explanations in food applications are rarely evaluated formally (Arrighi et al., 2026). When a segmentation of the product is available, the share of attribution falling on it provides an additional, task-specific check.

3. Materials and methods

Figure 1 gives an overview of the approach. The dataset is first audited for near-duplicate images and repeated specimens and split by specimen (Section 3.2), and the meat region of every image is isolated (Section 3.2.4). MeatGuard-S (Section 3.3) is then trained by distillation from a larger teacher (Section 3.4) and compressed for deployment on Android smartphones (Section 3.5), its explanations are evaluated quantitatively (Section 3.6), and all models are compared on the held-out test specimens (Section 3.7).

Figure 1

Overview of the MeatGuard pipeline



3.1 Dataset

MeatScan is a public dataset of RGB images of fresh and spoiled cow meat captured with low-cost smartphones in open-air markets, butcher shops and cold rooms in Ghana, without controlled lighting or backgrounds, and labelled from observable cues of texture, colour and surface condition by trained data collectors (Gyening et al., 2025). The released version contains 10,518 images (5,627 fresh, 4,891 spoiled) at 224 × 224 pixels. Its metadata file does not match the released file names, so all grouping information was derived from the images themselves.

3.2 Dataset audit and leakage control protocol

3.2.1 Near-duplicate audit

Consecutively numbered images are frequently near-identical frames of one piece of meat. To quantify this, a 64-bit difference hash (dHash) was computed for every image, and, because some images are rotated or mirrored copies of others, the distance between images *i* and *j* was taken as the smallest Hamming distance between any of the eight rotations and reflections of *i* and the original *j* (details in Supplementary Text S1):

$$d(i,j) = \min \text{ over } g \in G \text{ of } H(h(g(x_i)), h(x_j)) \quad (1)$$

where h is the dHash, H the Hamming distance and G the set of eight rotations and reflections. As a control for chance collisions, the same distance was computed between images of different classes.

Under a random 70/15/15 split at image level, 26.9% of test images had an exact hash match ($d = 0$) in the training set and 67.9% a match at $d \leq 4$, whereas none of 360,000 random cross-class pairs, used as a control for chance collisions, had $d \leq 6$. A random split of MeatScan therefore places near-copies of most test images in the training data.

3.2.2 Capture sessions

Images of the same class were linked into capture sessions when they lay within ten positions of each other in file order at $d \leq 12$, or anywhere at $d \leq 6$ (rotated or reflected copies); sessions are the connected components of the resulting graph. The thresholds were set from the cross-class control, of which 0.06% of pairs matched at $d \leq 12$ and none at $d \leq 6$, and no session contains images of both classes.

The procedure produced 1,666 sessions, 1,012 of them single images. The largest contains 2,619 images, 53.5% of all spoiled images, mostly of one piece of meat in a black polythene bag. Assigning whole sessions to partitions removes the near-duplicates from the held-out partitions: no validation or test image had a training near-duplicate at $d \leq 6$. A perceptual hash, however, compares whole images and cannot recognise one piece of meat photographed at another distance, angle or position, so a split by capture session can still place the same specimen in several partitions.

3.2.3 Specimen-level split

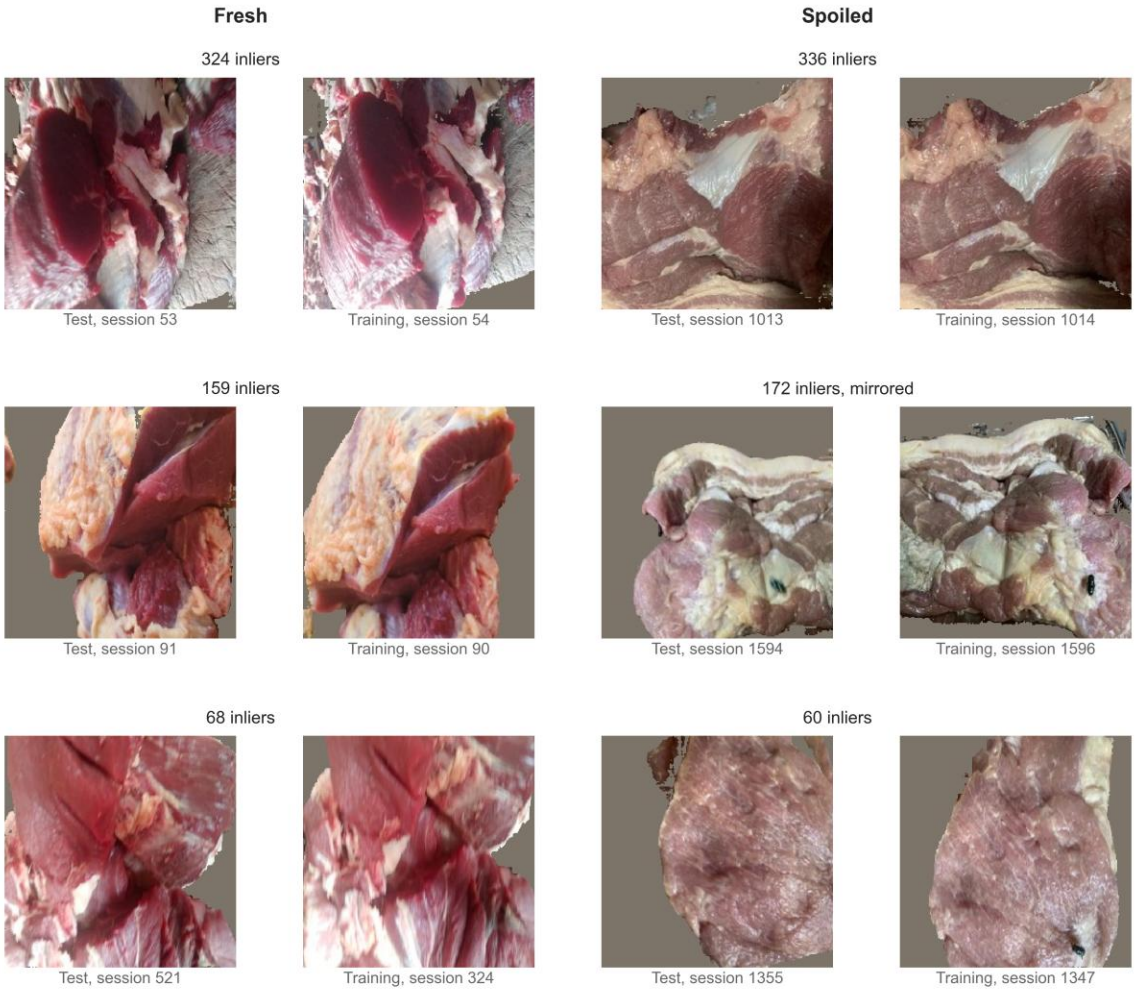
To detect the same piece of meat across sessions, candidate pairs of images were retrieved by the cosine similarity of global descriptors from the self-supervised vision transformer DINOv2 ViT-S/14 (Oquab et al., 2024), computed for every image and its mirror image, and verified

geometrically: AKAZE keypoints (Alcantarilla et al., 2013) were matched with the ratio test (Lowe, 2004), and a similarity transform was fitted by RANSAC (Fischler & Bolles, 1981). The number of inliers measures how much surface two images share. Images of the other class, which cannot show the same piece, served as a control for chance matches, and pairs known to show one piece as a positive control (Supplementary Text S1).

Applied to the split by capture session, this check found that 1,165 of the 1,578 test images (73.8%), from 254 of 318 test sessions, had a training image of the same class with at least 30 inliers among their three most similar training images; no image reached 20 inliers against the other-class control. Most matches showed the same piece re-photographed at another distance or angle, or a rotated or mirrored copy (Figure 2). A split by capture session therefore does not separate specimens in MeatScan.

Figure 2

Test and training images of the same pieces of meat under the split by capture session



Note. Each pair shows a test image and its best-matching training image of the same class in another capture session, under the split by capture session. The number above each pair gives the inliers of the geometric verification; "mirrored" marks a pair matched after mirroring one image. Pairs were selected per class at the 85th, 50th and 15th percentiles of the inlier count. Images are shown after meat-region cropping (Section 3.2.4).

Sessions were therefore linked into specimen groups: for each image, the six most similar images of the same class in other sessions (cosine similarity ≥ 0.7) were verified, and two sessions were linked when a pair reached 20 inliers. Whole groups were assigned to the training, validation and

test partitions with a 70/15/15 target per class, largest first, each to the partition furthest below its target; groups larger than a quarter of a held-out target were assigned to training, so that the held-out partitions consist of many specimens. Groups joined by a residual match between a held-out and a training image were merged and the assignment repeated, for four rounds (Supplementary Text S1).

The procedure produced 196 specimen groups, 82 fresh and 114 spoiled, 104 of which combine several capture sessions (Table 2). In the final split, one validation image and no test image had a training match of the same class with 20 or more inliers; at 15 inliers there were 11 validation (0.7%) and 20 test images (1.3%), against none in the other-class control. Measured by specimen group rather than by direct image match, 1,497 of the 1,578 test images of the split by capture session (94.9%) belonged to a group that also had images in its training partition, so the image-level figure of 73.8% is a lower bound. All models were trained and evaluated on the specimen-level split.

Table 2

Composition of the specimen-level split of MeatScan

Partition	Fresh images	Spoiled images	Total images	Fresh specimen groups	Spoiled specimen groups	Capture sessions
Training	3,939	3,423	7,362	34	52	559
Validation	844	734	1,578	24	31	543
Test	844	734	1,578	24	31	564
Total	5,627	4,891	10,518	82	114	1,666

Note. Every specimen group, and therefore every capture session, lies entirely within one partition. The two largest groups (1,676 fresh and 2,882 spoiled images) are in the training partition, which therefore contains fewer capture sessions than the held-out partitions; no held-out group contains more than 200 images.

3.2.4 Meat-region cropping

To prevent backgrounds such as polythene bags and metal trays from serving as cues to the label, the meat region of each image was isolated before training. The open-vocabulary detector Grounding DINO (S. Liu et al., 2025), prompted with the text "raw meat.", produced candidate boxes, which were passed as prompts to the segmentation model SAM 2.1 (Kirillov et al., 2023; Ravi et al., 2024); the union of the resulting masks was taken as the meat region. Pixels outside the mask were set to the ImageNet mean colour, which becomes zero after input normalisation, and each image was cropped to a square around the mask, enlarged by 8% on each side, and resized to 224 × 224 pixels. Thresholds, fallbacks and resolutions are given in Supplementary Text S1.

A box was detected in all 10,518 images, and one image needed a fallback. Visual inspection of 24 random masks per class found the meat captured in all 48, with small adjacent regions (part of a tray rim and a fold of polythene) in two spoiled-class masks. The median mask covered 89.5% of the frame for fresh and 29.7% for spoiled images, reflecting how much more of each spoiled photograph is background.

3.3 MeatGuard-S architecture

MeatGuard-S (Figure 3, Table 3) is a hybrid network in which convolutional stages, following the MobileNet family (Howard et al., 2017; Sandler et al., 2018), extract local colour and texture features, and a single transformer block, applied at a resolution where its cost is small, relates distant regions of the image.

Stage 1, a 3 × 3 convolution followed by a depthwise-separable convolution, both with stride 2, gives a 56 × 56 × 64 feature map. Stage 2 consists of three inverted residual blocks (Sandler et al., 2018) with an expansion ratio of 2, the first with stride 2, and outputs 28 × 28 × 128. Stage 3 projects the features to 256 channels, reduces the resolution to 14 × 14 by max pooling and applies one pre-norm transformer block to the resulting 196 tokens X:

$$X' = X + \mathrm{MHSA}(\mathrm{LN}(X)) \tag{2}$$

$$Y = X' + \mathrm{FFN}(\mathrm{LN}(X')) \tag{3}$$

where LN is layer normalisation, FFN is a two-layer feed-forward network (1 × 1 convolutions, 256 → 512 → 256, with SiLU activation) and MHSA is multi-head self-attention (Vaswani et al., 2017) with four heads of dimension $d_h = 64$:

$$\mathrm{Attention}(Q, K, V) = \mathrm{softmax}(QK^T / \sqrt{d_h})\, V \tag{4}$$

Stage 4 is an inverted residual block (256 → 384 channels) with squeeze-and-excitation channel attention (reduction ratio 4; Hu et al., 2018) and max pooling to 7 × 7, and the head applies global average pooling, dropout ($p = 0.3$) and a linear layer. All convolutions are followed by batch normalisation and, except for the linear projections, SiLU activation.

Table 3

MeatGuard-S stages, output sizes and cost for a 224 × 224 input

Stage	Operations	Output	Parameters (k)	MACs (M)
1	3 × 3 conv, stride 2; depthwise-separable conv, stride 2	56 × 56 × 64	3.5	18.2

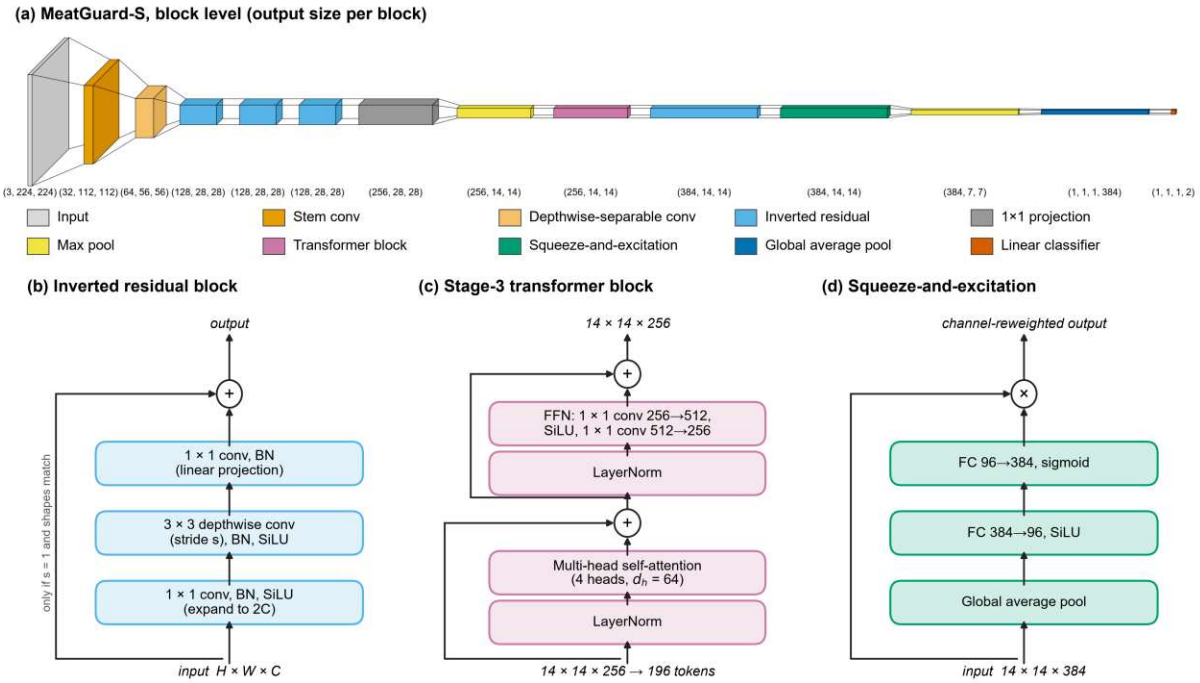
Stage	Operations	Output	Parameters (k)	MACs (M)
2	3 × inverted residual (expansion 2), first with stride 2	28 × 28 × 128	164.7	145.8
3	1 × 1 projection; 2 × 2 max pool; transformer block (4 heads)	14 × 14 × 256	559.6	148.1
4	Inverted residual (expansion 2); squeeze-and-excitation; 2 × 2 max pool	7 × 7 × 384	409.3	65.2
Head	Global average pooling; dropout 0.3; linear	2	0.8	< 0.1
Total			1,137.9	377.3

Note. MACs = multiply–accumulate operations, including the attention products (QK^T and the product with V). Floating-point operation counts reported elsewhere are often MACs under another name, so cross-paper comparisons of computational cost should be read with care.

The configuration was chosen by profiling variants on a central processing unit; it uses 36% fewer MACs and 25% fewer parameters than a variant with expansion ratio 4 and Stage 2 at 56 × 56.

Figure 3

The MeatGuard-S architecture



Note. (a) Block-level view of the network, rendered from the implemented model with VisualTorch (Hendria & Gavrikov, 2024); each box is one block, labelled with its output size (channels, height, width). (b) Inverted residual block, used in Stages 2 and 4; the residual connection is present only when the stride is 1 and the input and output shapes match. (c) Pre-norm transformer block of Stage 3, Eqs. (2)–(4). (d) Squeeze-and-excitation block of Stage 4.

3.4 Training and knowledge distillation

The teacher was selected from four ImageNet-pretrained architectures (Deng et al., 2009): EfficientNet-B0 (Tan & Le, 2019), ConvNeXt-Tiny (Z. Liu et al., 2022), Swin-T (Z. Liu et al., 2021) and ResNet-50 (He et al., 2016). Each was fine-tuned on the training partition with the same recipe (Table 4): AdamW (Loshchilov & Hutter, 2019), class-weighted cross-entropy with label smoothing, and early stopping on the validation macro-F1. The candidate with the highest validation macro-F1 was selected: ResNet-50 (0.916; Swin-T 0.907, ConvNeXt-Tiny 0.906, EfficientNet-B0 0.905), whose test macro-F1 was 0.912 (95% confidence interval from

resampled specimen groups, 0.846–0.950). The model of our earlier benchmark (Mettle et al., 2026) was not reused, because it was trained on a random image-level split and had seen the present test specimens.

All models used the same data augmentation, designed to mimic market photographs: flips, rotation of up to ±15°, colour jitter, Gaussian blur, a darkened strip simulating a shadow and a white overlay simulating haze (parameters in Supplementary Text S1). Validation and test images were only normalised with the ImageNet channel statistics.

MeatGuard-S was trained by knowledge distillation (Hinton et al., 2015) with feature alignment in the manner of hints (Romero et al., 2015). With z_s and z_t the student and teacher logits, σ the softmax and T the temperature, the loss is

$$L = \alpha\,\mathrm{CE}(y, z_s) + (1-\alpha)\,T^2\,\mathrm{KL}(\sigma(z_t/T) \,\|\, \sigma(z_s/T)) + \beta\,\|A(\bar{\varphi}_s) - \varphi_t\|^2 \quad (5)$$

where y is the label, $\bar{\varphi}_s$ the globally pooled Stage-3 output of the student, φ_t the penultimate features of the teacher and A a linear adaptor trained with the student (Supplementary Figure S1). The weights were $\alpha = 0.3$, $T = 4$ and $\beta = 0.1$, the teacher was kept frozen, and the student was trained for at most 150 epochs with the settings of Table 4; the checkpoint with the highest validation macro-F1 was kept.

3.5 Compression and on-device deployment

The distilled model was compressed by structured channel pruning: in each group of coupled layers, identified with a dependency graph (Fang et al., 2023), the 30% of channels with the smallest $\ell 1$-norms (Li et al., 2017) were removed, and the remaining counts were rounded to multiples of eight. The transformer block and the classifier output were excluded. Pruning

reduced the model from 1.14 to 0.85 million parameters and from 0.36 to 0.24 G multiply–accumulate operations (MACs), and the pruned model was fine-tuned for 10 epochs (Table 4).

For deployment, the model was exported through the Open Neural Network Exchange format to TensorFlow Lite and quantised to 8-bit integers after training (Jacob et al., 2018), with ranges calibrated on 500 training images. Because full-integer quantisation lost accuracy (Section 4.4), the quantisation error of each layer was measured and four alternatives were compared on the validation partition (Supplementary Text S2). The deployed model keeps the 22 operations of the two layer normalisations and the attention softmax in 32-bit floating point and all other weights and activations in 8-bit integers. Accuracy was evaluated on the test partition after each compression step.

The model was integrated into an offline Android application written in Kotlin, which runs it on the processor with the LiteRT 1.0.1 runtime and its XNNPACK kernels and repeats the preprocessing of the training pipeline. Because the application cannot yet isolate the meat on the device (Section 5.4), the user crops a photograph to a square. To check that the application reproduces the reference pipeline, all 1,578 test images were classified on the device, both from the stored crops and from the original photographs and their masks, and the outputs were compared with those of the TensorFlow Lite interpreter on a desktop computer. The test device was a mid-range HONOR tablet (NDL-W09; Qualcomm Snapdragon 680, 4 GB, Android 14). Latency was measured as the duration of the interpreter call for one 224 × 224 image, with one and four threads and with the Android Neural Networks API (NNAPI) delegate, over 200 timed inferences after 20 warm-up runs, repeated three times (Supplementary Text S2).

3.6 Explainability and evaluation of explanations

Three attribution methods were applied to the pruned and fine-tuned MeatGuard-S (before quantisation). Grad-CAM (Selvaraju et al., 2017) weights the feature maps A^k of a chosen layer by the spatially averaged gradient of the class score y^c and keeps the positive part:

$$\mathrm{L_Grad-CAM} = \mathrm{ReLU}(\Sigma_k \alpha_k A^k), \text{with } \alpha_k = (1 / Z) \Sigma_i \Sigma_j \partial y^c / \partial A^k{}_{ij} \quad (6)$$

where Z is the number of spatial positions. The layer, the output of Stage 4, was chosen among three candidates by the largest mean difference between the insertion and deletion areas defined below on 100 validation images, and the map was upsampled to the input size. SHAP values (Lundberg & Lee, 2017) were estimated with the expected-gradients explainer (50 training images as background, 100 samples per explanation) and averaged within SLIC superpixels (Achanta et al., 2012). Occlusion (Zeiler & Fergus, 2014) replaced 16 × 16-pixel patches, moved with a stride of 8 pixels, with the mean colour.

Because a plausible-looking map is not evidence that it reflects the model's behaviour (Adebayo et al., 2018), the explanations were evaluated with three measures. The first two are the deletion and insertion metrics of Petsiuk et al. (2018): pixels were replaced with the mean colour in decreasing order of attribution, in steps of 10% of the image, or restored in the same order to a mean-coloured image. With $f(t)$ the probability of the predicted class after a fraction t of the pixels has been changed, each curve was summarised by its normalised area,

$$\mathrm{AUC} = \int_0^1 f(t)\, dt \quad (7)$$

computed with the trapezoidal rule over the eleven points $t = 0, 0.1, \ldots, 1$. A faithful map yields a low deletion area and a high insertion area. Both were compared with the areas obtained from a random ranking of pixels.

The third measure uses the segmentation masks of Section 3.2.4 to test directly whether the model relies on the meat rather than on its surroundings. For an attribution map a and the meat mask M, transformed by the same crop as the image, the meat share is

$$S = \Sigma\{p \in M\} \max(a_p, 0) \,/\, \Sigma\{p\} \max(a_p, 0) \quad (8)$$

the fraction of positive attribution falling on the meat. It was compared with the fraction of the image covered by the mask, the value expected if attribution were spread uniformly. As a sanity check, the trained weights were randomised layer by layer, starting from the classifier (Adebayo et al., 2018), and the Spearman correlation between the original and the randomised maps was recorded. All measures were computed on a sample of 200 test images (100 per class, from 142 capture sessions in 35 specimen groups, at most two per session), and the randomisation check on 20 of them. Each method was compared with the random ranking, and the meat share with the mask area, by the Wilcoxon signed-rank test on paired per-image values (Wilcoxon, 1945), separately for fresh and spoiled predictions.

3.7 Experimental setup and evaluation

MeatGuard-S was compared with the teacher and with three widely used lightweight networks, MobileNetV3-Small (Howard et al., 2019), EfficientNet-B0 (Tan & Le, 2019) and ShuffleNet V2 (Ma et al., 2018), all initialised from ImageNet weights and fine-tuned with the teacher recipe on the same split and cropped images; MeatGuard-S trained without distillation served as an ablation. Compact models from the literature (Cao-Phan et al., 2026; Xu et al., 2026) were not re-implemented, because their published figures were obtained on other datasets and protocols.

Spoiled meat was treated as the positive class, because a spoiled sample passed as fresh is the costlier error. Models were evaluated on the test partition by accuracy, macro-averaged F1 score, spoiled-class recall and the area under the receiver operating characteristic curve (ROC

AUC). Because images of one specimen are not independent, 95% confidence intervals were obtained by resampling the 55 test specimen groups (2,000 resamples; Efron & Tibshirani, 1993), and paired differences in accuracy were tested with the cluster-adjusted McNemar test for matched pairs (Durkalski et al., 2003), with specimen groups as clusters, alongside the exact McNemar test over images (McNemar, 1947) and a bootstrap interval for the difference (10,000 resamples of specimen groups). As a reference for computational cost, the median latency of 100 single-image inferences was measured in PyTorch on one thread of a cloud x86-64 processor.

Cropping removes the background but keeps the outline and extent of the meat, and the mask covers much less of the frame in spoiled than in fresh photographs (Section 3.2.4). To test whether this extent carries information about the label, the mask area was used on its own as a score for the spoiled class, and the spoiled-class recall of each model was computed within quartiles of mask area. Table 4 lists the training settings.

Table 4

Training settings

Setting	**Teacher candidates and baselines**	**MeatGuard-S (distillation)**	**Pruned MeatGuard-S**
Initialisation	ImageNet weights	Random	Distilled model
Optimiser	AdamW, lr 3×10^{-4}, weight decay 0.05	AdamW, lr 10^{-3}, weight decay 10^{-4}	AdamW, lr 10^{-4}, weight decay 10^{-4}
Schedule	2-epoch warm-up, cosine decay	5-epoch warm-up, cosine decay	Cosine decay
Epochs	≤ 30, early stop after 8	≤ 150, early stop after 30	10

Setting	Teacher candidates and baselines	MeatGuard-S (distillation)	Pruned MeatGuard-S
Batch size	64	32	32
Loss	Class-weighted CE, label smoothing 0.1	Eq. (5), $\alpha = 0.3$, $T = 4$, $\beta = 0.1$	Class-weighted CE
Model selection	Validation macro-F1	Validation macro-F1, then validation loss	Validation macro-F1, then validation loss
Augmentation	As in Section 3.4	As in Section 3.4	As in Section 3.4

Note. CE = cross-entropy; lr = learning rate. Input size 224 × 224 for all models. The random seed was fixed at 42 for data splitting and training.

The models were implemented in PyTorch 2.11 with torchvision 0.26, pruned with Torch-Pruning 1.6, converted with onnx2tf 2.6.9 and TensorFlow 2.21.0, and trained on a single NVIDIA Tesla T4 GPU.

4. Results

4.1 Teacher selection and comparison of models

The four teacher candidates reached validation macro-F1 scores between 0.905 and 0.916, a margin smaller than the change between consecutive epochs (Supplementary Figure S3). Table 5 reports the test results of all models on the specimen-level split, together with the macro-F1 that the same recipes reached on the capture-session split. Under the specimen-level split, the test macro-F1 of the three ImageNet-pretrained lightweight networks fell from 0.994–0.996 to 0.901–0.929, from 7–10 to 112–156 misclassified test images, and that of MeatGuard-S fell by 7 to 12 points. Not all of this drop can be attributed to the removal of shared specimens, because the specimen-level split also changed the held-out set, which consists of groups of at most 200 images while the largest groups lie in training (Table 2). On the capture-session split, the

pretrained networks were almost as accurate on the 413 test images without a direct training match (0.983 to 0.988) as on the 1,165 with one (0.997 to 0.998), but most of these 413 images belonged to specimen groups with other images in training. Only 81 test images, 75 of them spoiled, came from groups without any training images; on these, the pretrained networks reached 0.95 to 0.98, too few and too unbalanced to separate the two effects.

MobileNetV3-Small and EfficientNet-B0 had the highest macro-F1 (0.929 and 0.927), above the teacher (0.912) and ShuffleNet V2 (0.901). The teacher had the highest spoiled-class recall (0.984) but classified 127 fresh images as spoiled. The confidence intervals spanned 0.10 to 0.17 in macro-F1, and the differences among the pretrained networks lie within them.

MeatGuard-S was less accurate. Trained without distillation, it reached a macro-F1 of 0.861. Distillation raised the validation macro-F1 from 0.850 to 0.867 and the spoiled-class recall from 0.835 to 0.905, but not the test macro-F1 (0.848), because it classified more fresh images as spoiled (170 against 97). The pruned and fine-tuned model reached 0.862 with 0.85 million parameters, the fewest of all models; directly after pruning it classified every validation image as spoiled, and one epoch of fine-tuning restored a validation macro-F1 of 0.860.

Table 5

Test results of all models

Model	Parameters (M)	MACs (G)	CPU latency (ms)	File size (MB)	Accuracy	Macro-F1 [95% CI]	Spoiled recall [95% CI]	ROC AUC	Macro-F1, session split
ResNet-50 (teacher)	23.51	4.12	119.4	90.0	0.912	0.912 [0.846, 0.950]	0.984 [0.962, 0.998]	0.990	0.995
EfficientNet-B0	4.01	0.40	35.7	15.6	0.928	0.927 [0.853, 0.969]	0.881 [0.743, 0.968]	0.981	0.996
MobileNetV3-Small	1.52	0.06	9.8	5.9	0.929	0.929 [0.852, 0.969]	0.907 [0.773, 0.992]	0.973	0.994
ShuffleNet V2	1.26	0.15	18.3	5.0	0.901	0.901 [0.811, 0.949]	0.931 [0.820, 0.986]	0.969	0.994
MeatGuard-S, without distillation	1.14	0.36	22.0	4.4	0.862	0.861 [0.751, 0.920]	0.835 [0.663, 0.934]	0.941	0.930
MeatGuard-S, distilled	1.14	0.36	23.3	4.4	0.848	0.848 [0.738, 0.915]	0.905 [0.791, 0.975]	0.944	0.966
MeatGuard-S, pruned and fine-tuned	0.85	0.24	19.3	3.3	0.862	0.862 [0.761, 0.921]	0.909 [0.784, 0.973]	0.952	0.971

Note. Test partition of 1,578 images from 55 specimen groups (564 capture sessions). CI = 95% confidence interval from 2,000 bootstrap resamples of test specimen groups. MACs = multiply–accumulate operations of the convolutional and linear layers for a 224 × 224 input. CPU latency is the median of 100 single-image inferences in PyTorch on one thread of a cloud x86-64 processor, given as a reference for relative cost only. File size is that of the 32-bit floating-point

weights. The last column gives the test macro-F1 of the same recipe on the capture-session split (1,578 test images from 318 sessions), which shared specimens between partitions; on that split, ConvNeXt-Tiny was selected as the teacher and distilled into MeatGuard-S, which was trained for at most 100 epochs with early stopping after 15. The quantised models are reported in Table 8.

The paired tests confirm that MeatGuard-S remained behind the pretrained networks (Table 6). The teacher and each pretrained network classified 145 to 180 test images correctly that the distilled MeatGuard-S misclassified, against 48 to 61 in the other direction ($p < 10^{-8}$ over images). With the tests clustered by specimen group, the accuracy differences of 0.064 to 0.081 remained significant for the teacher, EfficientNet-B0 and MobileNetV3-Small ($p = 0.009$ to 0.037), and the difference of 0.053 was borderline for ShuffleNet V2 ($p = 0.065$). Neither distillation nor pruning changed the accuracy significantly (clustered $p = 0.57$ and 0.32).

Table 6

Paired comparisons with the distilled MeatGuard-S on the test images

Compared model	Only distilled MeatGuard-S correct	Only compared model correct	Accuracy difference [95% CI]	*p*, images (exact McNemar)	*p*, clustered
ResNet-50 (teacher)	50	151	+0.064 [0.024, 0.117]	5.6×10^{-13}	0.009
EfficientNet-B0	54	180	+0.080 [0.016, 0.161]	5.0×10^{-17}	0.037
MobileNetV3-Small	48	176	+0.081 [0.015, 0.164]	2.3×10^{-18}	0.034
ShuffleNet V2	61	145	+0.053 [0.002, 0.112]	4.5×10^{-9}	0.065
MeatGuard-S, without distillation	104	126	+0.014 [−0.036, 0.063]	0.17	0.57

Compared model	Only distilled MeatGuard-S correct	Only compared model correct	Accuracy difference [95% CI]	*p*, images (exact McNemar)	*p*, clustered
MeatGuard-S, pruned and fine-tuned	50	73	+0.015 [−0.014, 0.043]	0.047	0.32

Note. Counts are numbers of the 1,578 test images classified correctly by one model of the pair only. Accuracy difference = accuracy of the compared model minus that of the distilled MeatGuard-S, with a 95% confidence interval from 10,000 bootstrap resamples of the 55 test specimen groups. Clustered p = cluster-adjusted McNemar test (Durkalski et al., 2003) with specimen groups as clusters.

Pruning gave the smallest model file, but MobileNetV3-Small required a quarter of the MACs of the pruned MeatGuard-S (0.06 against 0.24 G) and was about twice as fast on the reference processor (9.8 against 19.3 ms), despite having 1.8 times as many parameters.

4.2 Errors and the extent of the meat region

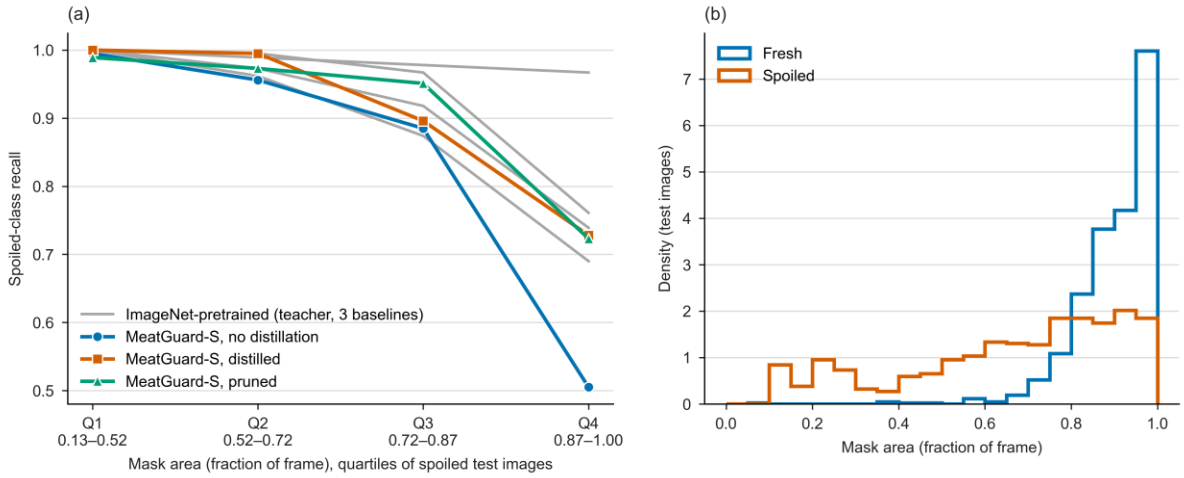
The 217 errors of the pruned MeatGuard-S came from 27 of the 55 test specimen groups; 150 were fresh images classified as spoiled, and a single fresh group of 96 images accounted for 36 of them. About half of the errors (111) were also made by at least one of the pretrained lightweight networks.

The extent of the meat region carried information about the label on its own: one minus the mask area separated spoiled from fresh test images with a ROC AUC of 0.82, because many spoiled photographs show a small piece of meat in a larger frame, whereas almost all fresh test photographs are close-ups (Figure 4b). The spoiled-class recall of every model except the teacher fell as the meat filled more of the frame (Figure 4a). In the two quartiles of spoiled test images with the smallest masks, every model reached a recall of at least 0.95; in the quartile with the largest masks (0.87 to 1.00 of the frame), the recall was 0.505 to 0.728 for the variants of

MeatGuard-S, 0.690 to 0.761 for the pretrained lightweight networks and 0.967 for the teacher. Across spoiled test images, the Spearman correlation between mask area and the predicted probability of spoilage was −0.60 to −0.83 for MeatGuard-S and −0.32 to −0.75 for the pretrained networks. On the capture-session split it had been −0.22 to 0.11 for the pretrained networks, so their dependence on mask area appeared only on unseen specimens.

Figure 4

Spoiled-class recall and the extent of the meat region



Note. (a) Spoiled-class recall of each model within quartiles of mask area, as a fraction of the original frame, over the 734 spoiled test images (about 183 per quartile). Grey lines show the four ImageNet-pretrained networks; the line that stays above 0.95 is the teacher. (b) Distribution of mask area for fresh and spoiled test images.

4.3 Evaluation of the explanations

For Grad-CAM, the output of Stage 4 gave the largest difference between the insertion and deletion areas on the validation sample (0.203, against 0.166 and 0.121 for the other two layers). On the 200-image test sample, the pruned MeatGuard-S had an accuracy of 0.815 (0.70 for fresh and 0.93 for spoiled images).

Table 7 and Figure 5 summarise the measures. Over the whole sample, all three methods ordered the pixels better than a random order by both criteria (Wilcoxon $p \leq 0.011$), occlusion best. The random baseline itself, however, differed sharply between the predicted classes (Figure 5a, b): replacing random pixels with the mean colour lowered the probability of a fresh prediction quickly (deletion area 0.416) but left spoiled predictions almost unchanged (0.921), and a partly restored mean-coloured image was likewise classified as spoiled (insertion area 0.424 for fresh and 0.914 for spoiled predictions). An image partly covered by the mean colour thus moves the model towards the spoiled class. In consequence, for fresh predictions all three methods were worse than the random order by deletion but better by insertion, whereas for spoiled predictions all three were better by deletion and only occlusion was better by insertion. The pooled scores therefore combine two opposite biases of the mean-colour baseline.

Table 7

Faithfulness and meat share of the explanations of the pruned MeatGuard-S

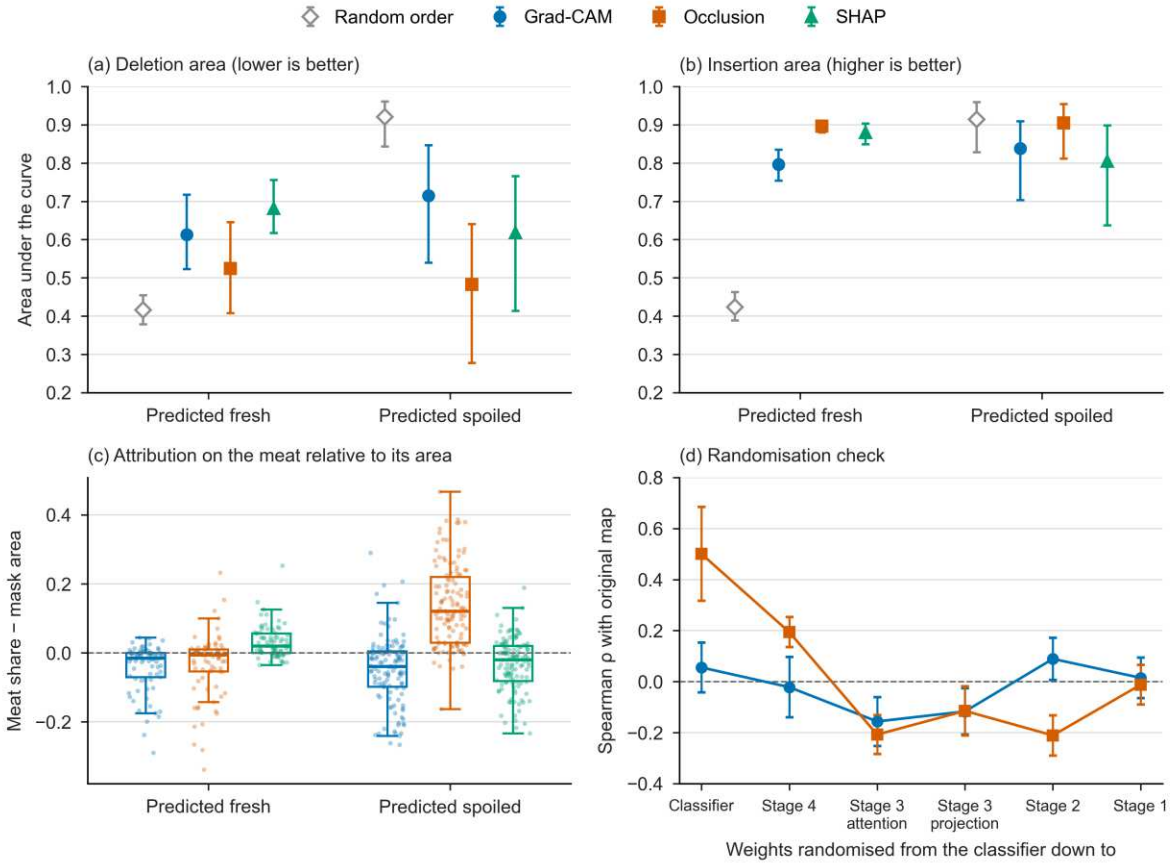
Predicted class	Method	Deletion AUC (lower is better)	Insertion AUC (higher is better)	Meat share
All (n = 200)	Random order / mask area	0.726	0.725	0.817
	Grad-CAM	0.675	0.822	0.771
	Occlusion	0.499	0.902	0.889
	SHAP	0.642	0.833	0.809
Fresh (n = 77)	Random order / mask area	0.416	0.424	0.926
	Grad-CAM	0.613	0.796	0.885

Predicted class	Method	Deletion AUC (lower is better)	Insertion AUC (higher is better)	Meat share
	Occlusion	0.524	0.896	0.893
	SHAP	0.681	0.880	0.957
Spoiled (*n* = 123)	Random order / mask area	0.921	0.914	0.749
	Grad-CAM	0.715	0.838	0.700
	Occlusion	0.483	0.905	0.886
	SHAP	0.618	0.804	0.717

Note. Means over the session-stratified test sample (100 images per true class from 142 capture sessions in 35 specimen groups), grouped by the class predicted by the model. For the reference rows, the deletion and insertion columns give the areas for a random pixel order and the meat-share column gives the mean fraction of the cropped image covered by the meat mask, the share expected if attribution were spread uniformly.

Figure 5

Faithfulness, meat share and randomisation check of the explanations of the pruned MeatGuard-S



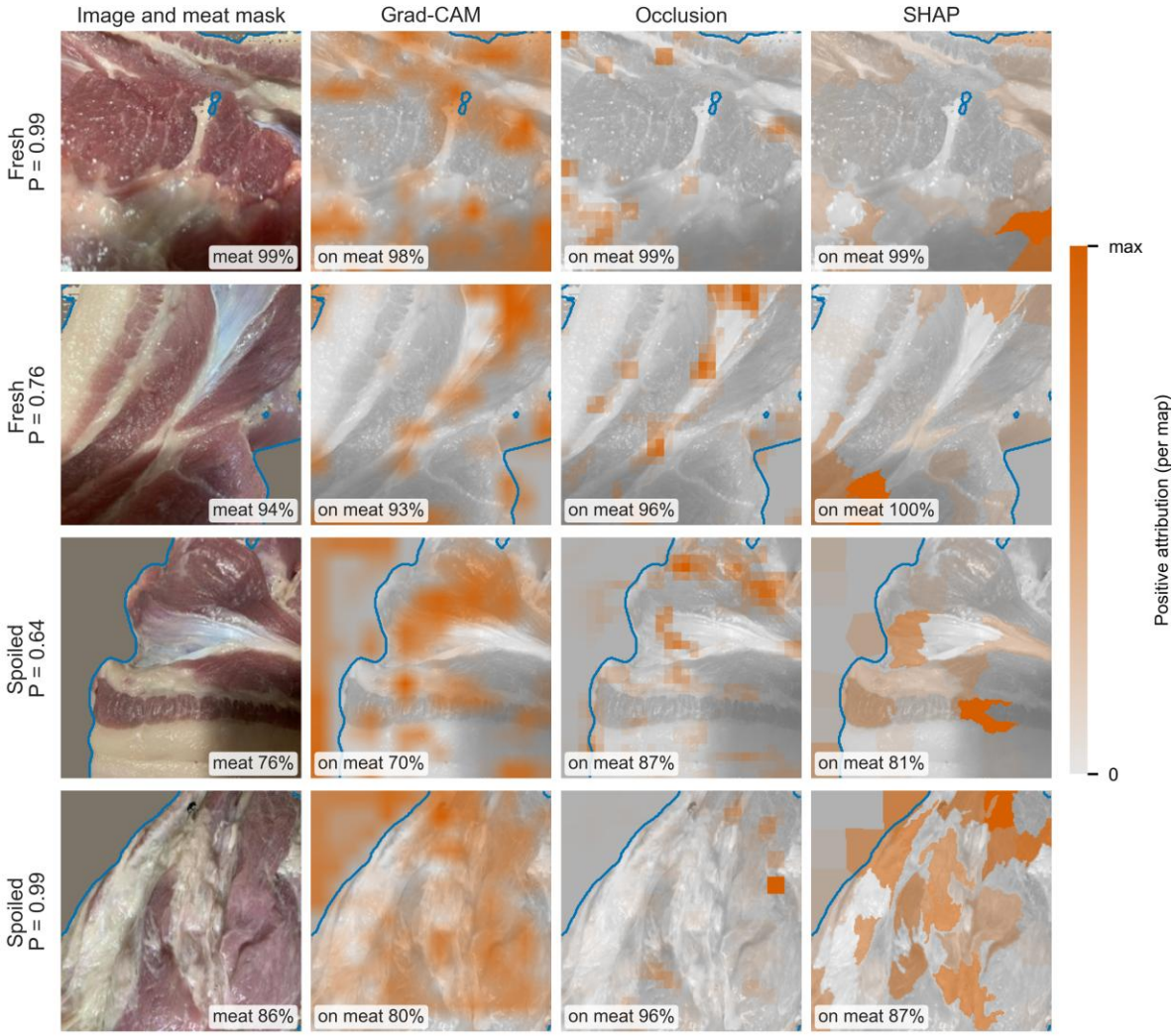
Note. (a, b) Mean deletion and insertion areas of each method and of a random pixel order, by predicted class (77 fresh and 123 spoiled predictions), with 95% intervals from 2,000 bootstrap resamples of specimen groups. (c) Meat share minus mask area per image; boxes show the median and quartiles, whiskers extend to 1.5 times the interquartile range, and values below zero mean less attribution on the meat than its area. (d) Mean Spearman correlation between the original map and the map after randomising the weights from the classifier down to the stage shown, over the 20 images of the randomisation check, with 95% intervals; SHAP was not included in this check.

The meat share separated the methods further (Figure 5c). For spoiled predictions, Grad-CAM and SHAP placed 70% and 72% of their positive attribution on meat that covered 75% of the image, less than its area (median paired differences −0.04 and −0.02; $p \leq 5 \times 10^{-5}$), whereas

occlusion placed 89% on it (+0.12). For fresh predictions, whose meat covered 93% of the image, the differences were small (−0.015 to +0.019). Per image, the meat shares of Grad-CAM and SHAP were closely related (Spearman $\rho = 0.89$), and those of occlusion less so (0.67 and 0.57). Figure 6 shows two typical images of each class: in the fresh close-ups all three methods placed 93 to 100% of their attribution on the meat, whereas in the spoiled examples Grad-CAM placed part of its attribution on the fill outside the meat and occlusion very little.

Figure 6

Explanations of MeatGuard-S for typical fresh and spoiled test images



Note. Each row is one test image from the explanation sample, with the predicted class and its probability. The first column shows the image with the outline of the meat mask (blue) and the share of the image covered by the mask; the other columns show the positive attribution of each method, scaled to its maximum within each map, and the share of that attribution falling on the meat. Images were selected per class as correctly classified images from different specimen groups whose Grad-CAM meat share minus mask area was closest to the class median.

The fresh images classified as spoiled differed little from the others: 30 of the 100 fresh images in the sample, from 11 of the 19 fresh specimen groups, with a median predicted probability of 0.79. Their meat covered slightly less of the image (median 0.91 against 0.94), and in the

misclassified images shown in Supplementary Figure S2 the methods placed 89 to 99% of their attribution on the meat, spread over the lean surface, so these errors were not explained by attribution to the fill.

In the randomisation check (Figure 5d), Grad-CAM depended on the learned weights: its mean correlation with the original map fell to 0.06 once the classifier had been randomised. Occlusion maps remained more similar (0.50 after randomising the classifier, mainly for fresh images) and decorrelated only once Stage 4 (0.19) and earlier stages were randomised.

4.4 Quantisation and on-device performance

Full-integer quantisation reduced the model file from 3.3 MB to 0.95 MB but did not preserve accuracy (Table 8): the test macro-F1 fell from 0.862 to 0.829 and the spoiled-class recall from 0.909 to 0.698, so that 222 spoiled images were classified as fresh, against 67 before quantisation. The ROC AUC fell less (0.952 to 0.941), so quantisation mainly shifted the outputs towards the fresh class rather than degrading the ranking of the images; the change in accuracy was not significant when clustered by specimen group ($p = 0.50$).

The loss came from the layer normalisations and the attention softmax of the transformer block. Keeping these 22 operations in floating point raised the validation macro-F1 from 0.786 to 0.861, close to the 0.863 obtained by keeping the whole transformer block in floating point at about twice the file size (1.8 MB); the other alternatives recovered nothing or could not be executed (Supplementary Text S2).

The deployed model has a file size of 0.95 MB and reached a test macro-F1 of 0.875 [0.779, 0.926] and a spoiled-class recall of 0.895 [0.765, 0.963]. Its apparent gain over the pruned floating-point model (32 against 12 discordant images; $p = 0.004$ over images, 0.13 clustered)

reversed on the validation partition (0.861 against 0.867) and reflects a slight shift of the outputs that suited a few test specimens rather than a gain from quantisation (Figure 7).

Table 8

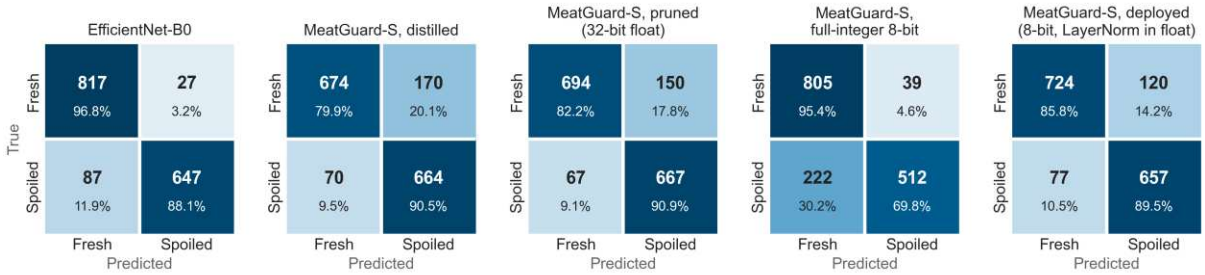
Quantisation of the pruned MeatGuard-S

Model	File size (MB)	Validation macro-F1	Test accuracy	Test macro-F1 [95% CI]	Test spoiled recall [95% CI]	Test ROC AUC
Pruned and fine-tuned, 32-bit floating point	3.3	0.867	0.862	0.862 [0.761, 0.921]	0.909 [0.784, 0.973]	0.952
Full-integer 8-bit	0.95	0.786	0.835	0.829 [0.697, 0.901]	0.698 [0.456, 0.840]	0.941
8-bit, transformer block in floating point	1.8	0.863	–	–	–	–
8-bit, eight largest-error layers in floating point	1.0	0.774	–	–	–	–
8-bit weights, 16-bit activations	1.0	not executable	–	–	–	–
8-bit, layer normalisation and softmax in floating point (deployed)	0.95	0.861	0.875	0.875 [0.779, 0.926]	0.895 [0.765, 0.963]	0.953

Note. The variant was chosen on the validation partition, so only the full-integer and the deployed models were evaluated on the test partition. CI = 95% confidence interval from 2,000 bootstrap resamples of test specimen groups. File size is that of the TensorFlow Lite file, or of the 32-bit weights for the pruned model.

Figure 7

Confusion matrices on the test partition, from the strongest pretrained baseline to the deployed MeatGuard-S



Note. Rows give the true class and columns the predicted class of the 1,578 test images (844 fresh, 734 spoiled). Each cell shows the number of images and their share of the true class; shading encodes the share on a scale common to all panels.

On the device, the application reproduced the reference pipeline. When it filled, cropped and resized the test photographs itself, its inputs were identical, pixel for pixel, to those of the reference pipeline, and the predicted label agreed for all 1,578 test images; the spoiled-class probabilities differed by at most 0.027, owing to the integer kernels of the runtime. The deployed model was, however, sensitive to small changes of the input: saving the crops as JPEG files at quality 95, which changed the pixel values by 0.7 grey levels on average, changed the predicted label of 73 test images (4.6%), 40 of them outside the band of 0.4 to 0.6 that the application marks as uncertain (Supplementary Text S2).

One inference of the deployed model took a median of 31.8 ms with one thread (interquartile range 0.1 ms) and 30.5 ms with four threads, which however produced a second cluster of inferences near 47 ms (Table 9); one thread therefore gave almost the same latency with far less variation. Loading the model took about 9 ms, and the NNAPI delegate could not be applied. Figure 8 shows the screens of the application.

Table 9

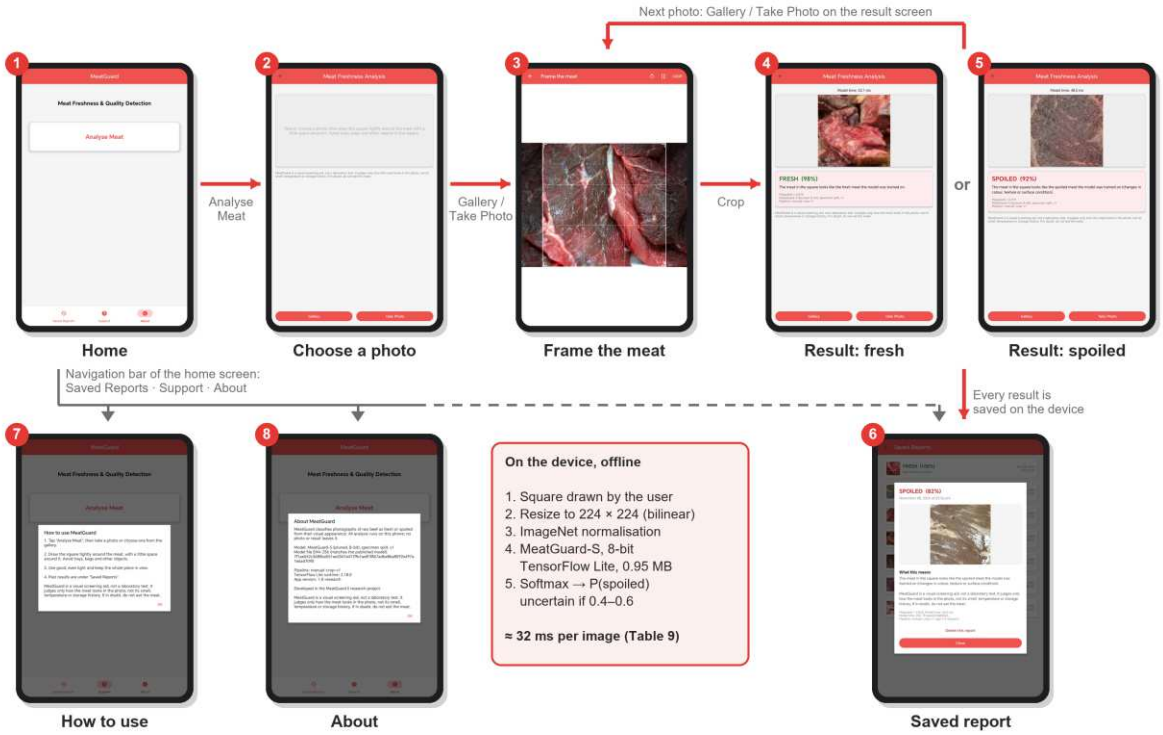
On-device latency of the deployed MeatGuard-S

Execution setting	Median (ms)	IQR (ms)	Mean ± SD (ms)	90th percentile (ms)	Maximum (ms)	Model loading (ms)
Processor, 1 thread (XNNPACK)	31.8	31.7–31.8	31.9 ± 0.7	31.9	41.9	8.8–9.6
Processor, 4 threads (XNNPACK)	30.5	30.2–46.9	35.8 ± 8.7	49.3	61.8	8.8–10.4
NNAPI delegate	could not be applied	–	–	–	–	–

Note. HONOR NDL-W09 tablet (Qualcomm Snapdragon 680, 4 GB, Android 14), LiteRT 1.0.1 (TensorFlow Lite 2.18.0), model file 0.95 MB. Times of the interpreter call for one 224 × 224 image, including the call from the application into the runtime, pooled over three repetitions of 200 timed inferences after 20 warm-up runs (600 per setting); the medians of the single repetitions were 31.7–31.8 ms with one thread and 30.4–30.5 ms with four threads. Model loading is the range over the three repetitions. The device was connected to power (battery 31–32%). The GPU delegate was not evaluated. IQR = interquartile range; SD = standard deviation.

Figure 8

Screens of the MeatGuard application on the test device, in the order of use



Note. Screenshots from the HONOR NDL-W09 tablet (application version 1.0-research), with the status bar removed. From the home screen (1), the user takes or chooses a photograph (2) and draws a square around the meat (3); the application then shows the predicted class with the probability of that class (4, 5), and probabilities of the spoiled class between 0.4 and 0.6 are reported as uncertain. Every result is stored on the device and can be reopened under Saved Reports (6). The navigation bar of the home screen also opens the instructions (7) and the model information, including the SHA-256 checksum of the model file (8). No photograph or result leaves the device.

5. Discussion

5.1 Pretrained models outperform models trained from scratch

With every specimen kept in one partition and the background removed, ImageNet-pretrained networks of 1.3 to 4.0 million parameters reached a test macro-F1 of 0.90 to 0.93, as high as a 24-million-parameter teacher, whereas MeatGuard-S, of similar size but trained without pretraining, remained 4 to 8 points behind; the difference was significant against three of the four pretrained networks in tests clustered by specimen group. Neither distillation nor pruning closed

the gap. Because no network was trained both with and without pretraining, these results show that the pretrained networks outperformed our scratch-trained hybrid on 7,362 training images from 86 specimen groups, not that pretraining as such outweighs architecture.

The comparison of cost points the same way. MeatGuard-S had the fewest parameters and the smallest file, but MobileNetV3-Small needed fewer MACs and less time per image; parameter and operation counts are both indirect measures of speed (Ma et al., 2018). Purpose-built compact freshness networks (Cao-Phan et al., 2026; Xu et al., 2026) should therefore be compared with pretrained lightweight networks on the same leakage-controlled split, with their cost measured on the target device. For an offline smartphone application on MeatScan-like images, the results favour a pretrained network such as MobileNetV3-Small.

5.2 Leakage survives deduplication

The largest effect in this study was that of the split. On the capture-session split, the pretrained networks misclassified fewer than 1% of the test images, close to the 100% accuracy reported on MeatScan with a random split (Faridi, 2026); on the specimen-level split, they misclassified 7 to 10%. The session split had removed all near-duplicates detectable by a perceptual hash, yet 74% of its test images had a training image of the same piece of meat, and 95% belonged to a specimen group with images in training. Deduplication alone therefore does not prevent leakage. Where collection records identify the specimens, they should define the split; where they do not, instance-level retrieval with geometric verification can recover them, at the cost of a few weak matches (1.3% of the test images at 15 inliers). Not all of the drop need be due to leakage, however: the specimen-level split also changed the held-out set, and the 81 session-split test images without any training images of their specimen are too few, and almost all spoiled, to apportion the drop between the two effects.

Separating specimens also made the evaluation less precise. With 55 test specimen groups, the confidence intervals spanned 0.10 to 0.17 in macro-F1, one fresh specimen accounted for a sixth of the errors of the pruned MeatGuard-S, and tests that treat images as independent overstated the evidence: two differences with $p < 0.05$ over images were not significant once clustered by specimen group. All specimens come from one collection campaign; how well these accuracies carry over to other markets, phones and storage conditions is not established by this study.

5.3 What the explanations and errors revealed

Over the whole sample, all three explanation methods ordered pixels better than random, but separately for each predicted class every method failed one of the two criteria, and in the randomisation check occlusion, unlike Grad-CAM, remained partly similar after the classifier had been randomised. A method that passes one check can fail another, and a pooled score can hide a failure within a class, which supports reporting several measures, per class, rather than example maps alone (Adebayo et al., 2018; Arrighi et al., 2026).

The meat share pointed to a cue that the cropping had not removed. For spoiled predictions, Grad-CAM and SHAP placed less attribution on the meat than its area would imply, so the model drew on the surrounding region, which in the cropped images is uniform fill and can convey only the outline and extent of the meat. The error analysis supports this reading: mask area alone predicted the label with a ROC AUC of 0.82, and every model except the teacher missed spoiled meat mainly when the meat filled the frame, as it does in almost all fresh photographs. The models thus appear to have learned, in part, a shortcut (Geirhos et al., 2020): small, isolated pieces of meat are more often spoiled in MeatScan. Because each crop is resized to 224 pixels, small masks are also enlarged more, so the apparent scale of the texture may contribute to the cue. On the capture-session split, the dependence was visible only for MeatGuard-S, so

recognising a specimen seen in training can mask the cues a model falls back on when the specimen is new. The evidence is correlational; testing the models on spoiled images cropped more tightly or with the fill replaced remains to be done.

The disagreement between the methods has a mechanical explanation. Occlusion and the deletion and insertion measures replace pixels with the mean colour, which is the colour of the fill. Occluding the fill changes nothing, so occlusion cannot attribute a prediction to it, and deletion and insertion cannot reward a map for doing so. The perturbation is not neutral either: covering parts of the meat with fill makes it look smaller and more broken up, like the spoiled images of MeatScan, which explains why random deletion removed fresh predictions quickly but left spoiled ones almost untouched, and why the faithfulness verdicts reversed between the classes. When a perturbation baseline coincides with a feature of the input, explanations and faithfulness measures built on that baseline are blind to the feature or biased by it; a segmentation-based measure such as the meat share, read together with an error analysis, can expose what they miss.

Background removal, the usual remedy for background bias in freshness studies (Bramantyo et al., 2026; Yu et al., 2025), is therefore necessary but not sufficient: the shape and size of what remains can carry the label. Cropping to a fixed scale rather than to the mask, varying the fill during training, or collecting more spoiled close-ups and fresh photographs taken from a distance are possible counter-measures.

5.4 Limitations

The study has several limitations. First, it uses one dataset from one country; the specimen-level split controls leakage within MeatScan but cannot measure transfer to other conditions. Second, the specimen groups were identified automatically: pieces photographed so differently that no

surface could be matched would not be linked, and 20 test images (1.3%) kept a weak match with a training image. Third, the labels reflect visual cues judged by the data collectors, not microbiological or chemical measurements, so the models learn the annotators' visual judgement. Fourth, each model was trained once with a fixed seed, and with 55 test specimen groups only the differences between MeatGuard-S and three of the pretrained networks were resolved. Fifth, the explanation measures were computed for MeatGuard-S only, on 200 test images, and their per-image tests do not account for clustering by specimen. Sixth, on-device latency was measured on one tablet and for the model call alone, and the deployed model changed 4.6% of its predictions under JPEG compression, so camera noise and compression on other phones may change predictions further. Finally, the application does not yet isolate the meat on the device, so the accuracies reported here apply to images prepared as in training, and the masks were inspected on a sample only.

6. Conclusions

On MeatScan, a split by capture session that removed all near-duplicate frames still left a training image of the same piece of meat for 74% of the test images (95% when counted by specimen group), and pretrained lightweight networks reached a test macro-F1 of 0.994 to 0.996 on it. On a split by specimen, built with instance-level retrieval and geometric verification and combined with background removal, the same networks reached 0.901 to 0.929. MeatGuard-S, trained without pretraining, reached 0.848 to 0.862; it was not improved by distillation, kept its accuracy when pruned and quantised to a 0.95 MB file that runs in 32 ms on an Android device, and remained less accurate than the pretrained networks and not cheaper to run than MobileNetV3-Small.

After cropping, the extent of the meat region still carries information about the label, and on unseen specimens all models lost spoiled-class recall when the meat filled the frame. Grad-CAM and SHAP pointed to the same cue, whereas occlusion and the deletion and insertion measures, which fill removed pixels with the mean colour, could not detect it and were biased by it.

Three recommendations follow: split freshness datasets by specimen and verify the split with instance-level matching; compare purpose-built compact networks with pretrained lightweight baselines under the same split, with cost measured on the target device; and evaluate explanations quantitatively, checking whether the perturbation baseline coincides with a feature of the input. Future work will test the shape cue directly, train a compact segmentation network so that the application can isolate the meat on the device, extend the evaluation to other markets and phones, and assess the application in the field with food inspectors.

Statements and Declarations

Funding

The authors declare that no funds, grants, or other support were received during the preparation of this manuscript.

Competing interests

The authors have no relevant financial or non-financial interests to disclose.

Author contributions

Samuel Opoku Berchie: Conceptualization; Data curation; Formal analysis; Methodology; Software; Writing – original draft. Peter Appiahene: Conceptualization; Data curation; Supervision; Writing – review & editing. Rose-Mary Owusuaa Mensah Gyening: Data curation; Supervision; Writing – review & editing. Henry Nii-Armah Mettle: Data curation; Software; Writing – review & editing. Benedicta Nyarko Anin-Acquah: Data curation; Writing – review & editing. Priscilla Akoto Peprah: Data curation; Validation. James Quayson: Data curation; Software. Emmanuel Botchway, Philip Sarfo-Manu, James Dorwodoe and Emmanuel Kwame Amoako: Software; Validation. All authors read and approved the final manuscript.

Ethics approval and consent

Not applicable. The study used a publicly available image dataset of cow meat offered for sale; no human participants or live animals were involved.

Data availability

The MeatScan images analysed in this study are publicly available on Zenodo (Gyening et al., 2025; https://doi.org/10.5281/zenodo.16764338). The specimen-level and capture-session splits, the segmentation masks and the trained weights are available on GitHub at https://github.com/Berchie-Sam/MeatGuard_II.

Code availability

The training and evaluation code, the trained and compressed weights and the Android application are available on GitHub at https://github.com/Berchie-Sam/MeatGuard_II.

Supplementary Information

The online version contains supplementary material: Text S1 (details of the dataset audit, specimen-level split, segmentation and augmentation), Text S2 (quantisation variants and on-device checks), Figure S1 (connection of the MeatGuard-S building blocks to the teacher and the loss), Figure S2 (explanations of fresh test images classified as spoiled) and Figure S3 (training curves).

References

Abd Elfattah, M., Ewees, A. A., Darwish, A., & Hassanien, A. E. (2025). Detection and classification of meat freshness using an optimized deep learning method. *Food Chemistry, 489*, 144783. https://doi.org/10.1016/j.foodchem.2025.144783

Achanta, R., Shaji, A., Smith, K., Lucchi, A., Fua, P., & Süsstrunk, S. (2012). SLIC superpixels compared to state-of-the-art superpixel methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 34*(11), 2274–2282. https://doi.org/10.1109/tpami.2012.120

Adebayo, J., Gilmer, J., Muelly, M., Goodfellow, I., Hardt, M., & Kim, B. (2018). Sanity checks for saliency maps. In *Advances in Neural Information Processing Systems* (Vol. 31, pp. 9505–9515). Curran Associates.

Adesola, R. O., Hossain, D., Ogundijo, O. A., Idris, I., Hamzat, A., Gulumbe, B. H., Bakre, A. A., Banwo, O. G., & Lucero-Prisno, D. E. (2024). Challenges, health risks and recommendations on meat handling practices in Africa: A comprehensive review.

Environmental Health Insights, 18, 11786302241301991. https://doi.org/10.1177/11786302241301991

Alcantarilla, P. F., Nuevo, J., & Bartoli, A. (2013). Fast explicit diffusion for accelerated features in nonlinear scale spaces. In *Proceedings of the British Machine Vision Conference* (pp. 13.1–13.11). BMVA Press. https://doi.org/10.5244/C.27.13

Arrighi, L., de Moraes, I. A., Zullich, M., Simonato, M., Barbin, D. F., & Barbon Junior, S. (2026). Explainable artificial intelligence techniques for interpretation of food models: A review. *Artificial Intelligence Review, 59*(8), 176. https://doi.org/10.1007/s10462-026-11553-2

Arsalane, A., Klilou, A., & Barbri, N. E. (2024). Performance evaluation of machine learning algorithms for meat freshness assessment. *International Journal of Electrical and Computer Engineering (IJECE), 14*(5), 5858. https://doi.org/10.11591/ijece.v14i5.pp5858-5865

Bhuiyan, Z. W., Haider, S. A. R., Haque, A., Uddin, M. R., & Hasan, M. (2024). IoT based meat freshness classification using deep learning. *IEEE Access, 12*, 196047–196069. https://doi.org/10.1109/access.2024.3520029

Bramantyo, H. A., Faridi, M. A., Chen, R., Harris, C., & Sun, Y. (2026). Deep learning-based meat freshness detection with segmentation and OOD-aware classification. *IEEE Access, 14*, 34939–34952. https://doi.org/10.1109/access.2026.3669074

Cao-Phan, K.-D., Dang, H.-K., Tran-Quynh, H.-T., Lam-Doan, M.-P., Luu, H.-T., & Bui, H.-Q. (2026). Fast and accurate meat freshness classification using depthwise separable

convolution and SPPF. *Food Analytical Methods, 19*(6), 180. https://doi.org/10.1007/s12161-026-03083-y

Chen, Y., Zhou, J., Oz, F., He, Z., Wang, Z., Chen, Q., Zeng, M., & Chen, J. (2026). Study on methods for measuring beef color and predicting storage time based on computer vision. *Meat Science, 234*, 110044. https://doi.org/10.1016/j.meatsci.2026.110044

Dave, D., & Ghaly, A. E. (2011). Meat spoilage mechanisms and preservation techniques: A critical review. *American Journal of Agricultural and Biological Sciences, 6*(4), 486–510. https://doi.org/10.3844/ajabssp.2011.486.510

Deliçay, N., Ozogul, Y., Mohammad Hamidy, S., Takadaş, F., & Kuvvetli, Y. (2026). Deep learning-based assessment of European seabass (Dicentrarchus labrax) fillet freshness and quality using image data. *Foods, 15*(19), 3385. https://doi.org/10.3390/foods15193385

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., & Fei-Fei, L. (2009). ImageNet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition* (pp. 248–255). https://doi.org/10.1109/cvpr.2009.5206848

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., & Houlsby, N. (2021). An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*. https://doi.org/10.48550/arXiv.2010.11929

Durkalski, V. L., Palesch, Y. Y., Lipsitz, S. R., & Rust, P. F. (2003). Analysis of clustered matched-pair data. *Statistics in Medicine, 22*(15), 2417–2428. https://doi.org/10.1002/sim.1438

Efron, B., & Tibshirani, R. J. (1993). *An introduction to the bootstrap*. Springer US. https://doi.org/10.1007/978-1-4899-4541-9

Fang, G., Ma, X., Song, M., Bi Mi, M., & Wang, X. (2023). DepGraph: Towards any structural pruning. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 16091–16101). https://doi.org/10.1109/cvpr52729.2023.01544

Faridi, A. A. (2026). Meat Scan v4: Automated meat freshness detection via high-resolution deep learning with ConvNeXt and multi-scale test-time augmentation. *Indian Journal of Computer Science and Technology, 5*(2), 427–436. https://doi.org/10.59256/indjcst.20260502048

Fischler, M. A., & Bolles, R. C. (1981). Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM, 24*(6), 381–395. https://doi.org/10.1145/358669.358692

Geirhos, R., Jacobsen, J.-H., Michaelis, C., Zemel, R., Brendel, W., Bethge, M., & Wichmann, F. A. (2020). Shortcut learning in deep neural networks. *Nature Machine Intelligence, 2*(11), 665–673. https://doi.org/10.1038/s42256-020-00257-z

Golder, S., Raha, A. D., Rubel, F. A., Adhikary, A., Bairagi, A. K., Debnath, R., & Gain, M. (2024). Lightweight and trustworthy deep learning model for fish freshness detection: An XAI-driven approach. In *2024 27th International Conference on Computer and Information Technology (ICCIT)* (pp. 2946–2951). https://doi.org/10.1109/iccit64611.2024.11022429

Gou, J., Yu, B., Maybank, S. J., & Tao, D. (2021). Knowledge distillation: A survey. *International Journal of Computer Vision, 129*(6), 1789–1819. https://doi.org/10.1007/s11263-021-01453-z

Grace, D. (2015). Food safety in low and middle income countries. *International Journal of Environmental Research and Public Health, 12*(9), 10490–10507. https://doi.org/10.3390/ijerph120910490

Gyening, R.-M. O. M., Akoto, M. A., Owusu-Agyemang, K., Amoako-Banning, L., Takyi, K., & Appiahene, P. (2025). MeatScan: An image dataset for machine learning-based classification of fresh and spoiled cow meat. *Data in Brief, 62*, 112045. https://doi.org/10.1016/j.dib.2025.112045

Hamidy, S. M., Kuvvetli, Y., Sakarya, Y., Özkütük, S. T., & Özoğul, Y. (2025). Deep learning-based prediction of fish freshness and purchasability using multi-angle image data. *Foods, 15*(1), 68. https://doi.org/10.3390/foods15010068

Han, S., Mao, H., & Dally, W. J. (2016). Deep compression: Compressing deep neural networks with pruning, trained quantization and Huffman coding. In *International Conference on Learning Representations*. https://doi.org/10.48550/arXiv.1510.00149

Han, Y., Jia, X., Yu, L., Ma, P., Zong, S., Sun, J., Hou, W., Yi, Y., Zhang, X., & Chen, S. (2026). Machine learning-based meat freshness evaluation: Principle, pipeline and application. *Critical Reviews in Food Science and Nutrition, 66*(21), 3913–3939. https://doi.org/10.1080/10408398.2026.2616384

Havelaar, A. H., Kirk, M. D., Torgerson, P. R., Gibb, H. J., Hald, T., Lake, R. J., Praet, N., Bellinger, D. C., de Silva, N. R., Gargouri, N., Speybroeck, N., Cawthorne, A., Mathers,

C., Stein, C., Angulo, F. J., & Devleesschauwer, B. (2015). World Health Organization global estimates and regional comparisons of the burden of foodborne disease in 2010. *PLOS Medicine, 12*(12), e1001923. https://doi.org/10.1371/journal.pmed.1001923

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 770–778). https://doi.org/10.1109/cvpr.2016.90

Hendria, W. F., & Gavrikov, P. (2024). VisualTorch: Streamlining visualization for PyTorch neural network architectures. *Journal of Open Source Software, 9*(102), 6678. https://doi.org/10.21105/joss.06678

Hidalgo, M. M., Lima, R. C., De Nadai Fernandes, E. A., Bacchi, M. A., & Sarriés, G. A. (2025). Leveraging pre-trained computer vision models for accurate classification of meat freshness. *Food Chemistry, 495*, 146430. https://doi.org/10.1016/j.foodchem.2025.146430

Hinton, G., Vinyals, O., & Dean, J. (2015). Distilling the knowledge in a neural network. *arXiv*. https://doi.org/10.48550/arXiv.1503.02531

Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., & Adam, H. (2017). MobileNets: Efficient convolutional neural networks for mobile vision applications. *arXiv*. https://doi.org/10.48550/arXiv.1704.04861

Howard, A., Sandler, M., Chen, B., Wang, W., Chen, L.-C., Tan, M., Chu, G., Vasudevan, V., Zhu, Y., Pang, R., Adam, H., & Le, Q. (2019). Searching for MobileNetV3. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)* (pp. 1314–1324). https://doi.org/10.1109/iccv.2019.00140

907 Hu, J., Shen, L., & Sun, G. (2018). Squeeze-and-excitation networks. In *2018 IEEE/CVF*
908 *Conference on Computer Vision and Pattern Recognition* (pp. 7132–7141).
909 https://doi.org/10.1109/cvpr.2018.00745

910 Jacob, B., Kligys, S., Chen, B., Zhu, M., Tang, M., Howard, A., Adam, H., & Kalenichenko, D.
911 (2018). Quantization and training of neural networks for efficient integer-arithmetic-only
912 inference. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*
913 (pp. 2704–2713). https://doi.org/10.1109/cvpr.2018.00286

914 Jian, R., Li, G., Jun, X., & Shi, G. (2025). Nondestructive freshness recognition of chicken breast
915 meat based on deep learning. *Scientific Reports, 15*(1), 27538.
916 https://doi.org/10.1038/s41598-025-13576-1

917 Kalinowska, K., Wojnowski, W., & Tobiszewski, M. (2021). Smartphones as tools for equitable
918 food quality assessment. *Trends in Food Science & Technology, 111*, 271–279.
919 https://doi.org/10.1016/j.tifs.2021.02.068

920 Kang, Y., Li, Y., Wang, H., Guo, J., Huang, Z., & Du, J. (2025). The use of deep learning in
921 primary agricultural products freshness assessment: A systematic review. *Journal of Food*
922 *Science, 90*(9), e70535. https://doi.org/10.1111/1750-3841.70535

923 Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S.,
924 Berg, A. C., Lo, W.-Y., Dollár, P., & Girshick, R. (2023). Segment anything. In
925 *Proceedings of the IEEE/CVF International Conference on Computer Vision* (pp. 4015–
926 4026). https://doi.org/10.1109/ICCV51070.2023.00371

Kozan, H. I., & Akyürek, H. A. (2024). Development of a mobile application for rapid detection of meat freshness using deep learning. *Theory and Practice of Meat Processing, 9*(3), 249–257. https://doi.org/10.21323/2414-438x-2024-9-3-249-257

Külcü, S., & Balpetek Külcü, D. (2026). Bias mitigation in ensemble-based meat freshness classification using Grad-CAM. *Balkan Journal of Electrical and Computer Engineering, 14*, 74–82. https://doi.org/10.17694/bajece.1817907

Li, H., Kadav, A., Durdanovic, I., Samet, H., & Graf, H. P. (2017). Pruning filters for efficient ConvNets. In *International Conference on Learning Representations*. https://doi.org/10.48550/arXiv.1608.08710

Li, Y., Yuan, G., Wen, Y., Hu, J., Evangelidis, G., Tulyakov, S., Wang, Y., & Ren, J. (2022). EfficientFormer: Vision transformers at MobileNet speed. In *Advances in Neural Information Processing Systems 35* (pp. 12934–12949). https://doi.org/10.52202/068431-0940

Lin, H.-D., Chen, J.-L., & Lin, C.-H. (2025). A deep learning-based sensing system for identifying salmon and rainbow trout meat and grading freshness for consumer protection. *Sensors, 25*(20), 6299. https://doi.org/10.3390/s25206299

Liu, C., Zhang, J., Chen, K., & Huang, J. (2025). BBSNet: An intelligent grading method for pork freshness based on few-shot learning. *Foods, 14*(14), 2480. https://doi.org/10.3390/foods14142480

Liu, S., Zeng, Z., Ren, T., Li, F., Zhang, H., Yang, J., Jiang, Q., Li, C., Yang, J., Su, H., Zhu, J., & Zhang, L. (2025). Grounding DINO: Marrying DINO with grounded pre-training for

open-set object detection. In *Computer Vision – ECCV 2024* (pp. 38–55). Springer Nature Switzerland. https://doi.org/10.1007/978-3-031-72970-6_3

Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., & Guo, B. (2021). Swin Transformer: Hierarchical vision transformer using shifted windows. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)* (pp. 9992–10002). https://doi.org/10.1109/iccv48922.2021.00986

Liu, Z., Mao, H., Wu, C.-Y., Feichtenhofer, C., Darrell, T., & Xie, S. (2022). A ConvNet for the 2020s. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 11966–11976). https://doi.org/10.1109/cvpr52688.2022.01167

Loshchilov, I., & Hutter, F. (2019). Decoupled weight decay regularization. In *International Conference on Learning Representations*. https://doi.org/10.48550/arXiv.1711.05101

Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision, 60*(2), 91–110. https://doi.org/10.1023/B:VISI.0000029664.99615.94

Lundberg, S. M., & Lee, S.-I. (2017). A unified approach to interpreting model predictions. In *Advances in Neural Information Processing Systems* (Vol. 30, pp. 4765–4774). Curran Associates.

Ma, N., Zhang, X., Zheng, H.-T., & Sun, J. (2018). ShuffleNet V2: Practical guidelines for efficient CNN architecture design. In *Computer Vision – ECCV 2018* (pp. 122–138). Springer International Publishing. https://doi.org/10.1007/978-3-030-01264-9_8

Maaz, M., Shaker, A., Cholakkal, H., Khan, S., Zamir, S. W., Anwer, R. M., & Shahbaz Khan, F. (2023). EdgeNeXt: Efficiently amalgamated CNN-transformer architecture for mobile vision applications. In *Computer Vision – ECCV 2022 Workshops* (pp. 3–20). Springer Nature Switzerland. https://doi.org/10.1007/978-3-031-25082-8_1

McNemar, Q. (1947). Note on the sampling error of the difference between correlated proportions or percentages. *Psychometrika, 12*(2), 153–157. https://doi.org/10.1007/bf02295996

Mehta, S., & Rastegari, M. (2022). MobileViT: Light-weight, general-purpose, and mobile-friendly vision transformer. In *International Conference on Learning Representations*. https://doi.org/10.48550/arXiv.2110.02178

Mettle, H. N.-A., Appiahene, P., Botchway, E., Quayson, J., Gyening, R.-M. O. M., Sarfo-Manu, P., Dorwodoe, J., Anin-Acquah, B. N., Amoako, E. K., Peprah, P. A., & Berchie, S. O. (2026). Benchmark study: CNN vs. vision transformer architectures for RGB meat freshness classification [Manuscript submitted for publication].

Modzelewska-Kapituła, M., & Jun, S. (2022). The application of computer vision systems in meat science and industry – a review. *Meat Science, 192*, 108904. https://doi.org/10.1016/j.meatsci.2022.108904

Nychas, G.-J. E., Skandamis, P. N., Tassou, C. C., & Koutsoumanis, K. P. (2008). Meat spoilage during distribution. *Meat Science, 78*(1-2), 77–89. https://doi.org/10.1016/j.meatsci.2007.06.020

Oquab, M., Darcet, T., Moutakanni, T., Vo, H., Szafraniec, M., Khalidov, V., Fernandez, P., Haziza, D., Massa, F., El-Nouby, A., Assran, M., Ballas, N., Galuba, W., Howes, R.,

Huang, P.-Y., Li, S.-W., Misra, I., Rabbat, M., Sharma, V., … Bojanowski, P. (2024). DINOv2: Learning robust visual features without supervision. *Transactions on Machine Learning Research*. https://arxiv.org/abs/2304.07193

Petsiuk, V., Das, A., & Saenko, K. (2018). RISE: Randomized input sampling for explanation of black-box models. In *Proceedings of the British Machine Vision Conference*. https://doi.org/10.48550/arXiv.1806.07421

Ravi, N., Gabeur, V., Hu, Y.-T., Hu, R., Ryali, C., Ma, T., Khedr, H., Rädle, R., Rolland, C., Gustafson, L., Mintun, E., Pan, J., Alwala, K. V., Carion, N., Wu, C.-Y., Girshick, R., Dollár, P., & Feichtenhofer, C. (2024). SAM 2: Segment anything in images and videos. *arXiv*. https://doi.org/10.48550/arXiv.2408.00714

Ribeiro, M., Singh, S., & Guestrin, C. (2016). "why should i trust you?": Explaining the predictions of any classifier. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations* (pp. 97–101). https://doi.org/10.18653/v1/n16-3020

Romero, A., Ballas, N., Kahou, S. E., Chassang, A., Gatta, C., & Bengio, Y. (2015). FitNets: Hints for thin deep nets. In *International Conference on Learning Representations*. https://doi.org/10.48550/arXiv.1412.6550

Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., & Chen, L.-C. (2018). MobileNetV2: Inverted residuals and linear bottlenecks. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 4510–4520). https://doi.org/10.1109/cvpr.2018.00474

Savarolyia, M., Moradi Marnilo, S., Amiri Samani, S., Jafarian, M., & Taghinia, P. (2026). Smartphone-assisted technologies for real-time monitoring of meat, seafood, and dairy products freshness – a critical discussion and comparative analysis. *Food Reviews International, 42*(7), 4789–4810. https://doi.org/10.1080/87559129.2026.2637631

Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., & Batra, D. (2017). Grad-CAM: Visual explanations from deep networks via gradient-based localization. In *2017 IEEE International Conference on Computer Vision (ICCV)* (pp. 618–626). https://doi.org/10.1109/iccv.2017.74

Sherin, Z., Bashir, O., Mondal, I. H., & Pawase, P. A. (2026). Advances in hyperspectral imaging for spoilage detection in meat and poultry: A non-invasive approach. *Trends in Food Science & Technology, 169*, 105558. https://doi.org/10.1016/j.tifs.2026.105558

Song, S., Guo, Q., Duan, X., Shi, X., & Liu, Z. (2024). Research on pork cut and freshness determination method based on computer vision. *Foods, 13*(24), 3986. https://doi.org/10.3390/foods13243986

Taheri-Garavand, A., Fatahi, S., Banan, A., & Makino, Y. (2019). Real-time nondestructive monitoring of common carp fish freshness using robust vision-based intelligent modeling approaches. *Computers and Electronics in Agriculture, 159*, 16–27. https://doi.org/10.1016/j.compag.2019.02.023

Taheri-Garavand, A., Nasiri, A., Banan, A., & Zhang, Y.-D. (2020). Smart deep learning-based approach for non-destructive freshness diagnosis of common carp fish. *Journal of Food Engineering, 278*, 109930. https://doi.org/10.1016/j.jfoodeng.2020.109930

Tan, M., & Le, Q. V. (2019). EfficientNet: Rethinking model scaling for convolutional neural networks. In *Proceedings of the 36th International Conference on Machine Learning* (pp. 6105–6114). PMLR.

Tsagkatakis, G., Roumpakis, S., Nikolidakis, S., Petra, E., Mantes, A., Kapantagakis, A., Grigorakis, K., Katselis, G., Vlahos, N., & Tsakalides, P. (2021). Knowledge distillation from multispectral images for fish freshness estimation. *Electronic Imaging, 33*(12), 27–1–27–7. https://doi.org/10.2352/issn.2470-1173.2021.12.fais-027

Ulucan, O., Karakaya, D., & Turkan, M. (2019). Meat quality assessment based on deep learning. In *2019 Innovations in Intelligent Systems and Applications Conference (ASYU)* (pp. 1–5). https://doi.org/10.1109/asyu48272.2019.8946388

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. In *Advances in Neural Information Processing Systems* (Vol. 30, pp. 5998–6008). Curran Associates. https://doi.org/10.48550/arXiv.1706.03762

Wilcoxon, F. (1945). Individual comparisons by ranking methods. *Biometrics Bulletin, 1*(6), 80–83. https://doi.org/10.2307/3001968

Xu, W., He, Y., Que, L., Tang, C., Li, J., Zhang, C., Dong, Q., Sheng, H., Liu, P., Wang, W., & Liu, D. (2026). RT-BFE network: A lightweight deep learning model for real-time beef freshness evaluation and edge devices deployment. *Computers and Electronics in Agriculture, 250*, 111905. https://doi.org/10.1016/j.compag.2026.111905

Yu, X., Xu, Y., Qu, C., Guo, S., Jiang, S., Chen, L., & Zhou, Y. (2025). Research on an intelligent grading method for beef freshness in complex backgrounds based on the DEVA-ConvNeXt model. *Foods, 14*(24), 4178. https://doi.org/10.3390/foods14244178

Zeiler, M. D., & Fergus, R. (2014). Visualizing and understanding convolutional networks. In *Computer Vision – ECCV 2014* (pp. 818–833). Springer. https://doi.org/10.1007/978-3-319-10590-1_53

Zhou, C., Pi, J., Chen, X., Wang, D., & Liu, J. (2025). Identification and analysis of pork freshness quality based on improved MobileNetV3. *Applied Engineering in Agriculture, 41*(1), 57–66. https://doi.org/10.13031/aea.16131

Supplementary Files

This is a list of supplementary files associated with this preprint. Click to download.

- Supplementarymaterial.docx