Contact-Free Coffee Weight Estimation using Scanline Approach

Pratyush Paliwal
TU Darmstadt
pratyush.paliwal@stud.tu-darmstadt.de

Sankalp Tripathi
TU Darmstadt
sankalp.tripathi@stud.tu-darmstadt.de

Himanshi Agrawal
TU Darmstadt
himanshi.agrawal@stud.tu-darmstadt.de

Abstract

Estimation of cumulative coffee weight from visual data offers a non-intrusive alternative to traditional grind-by-weight systems, which are often susceptible to environmental disturbances such as vibrations. However, this task remains challenging, as it requires capturing subtle visual cues and fine-grained temporal dynamics from high-speed video streams of falling coffee particles. This paper proposes a scanline-based representation that transforms a video sequence into a single structured image by vertically stacking temporally ordered horizontal slices extracted from consecutive frames. This representation aims to encode temporal information directly into a spatial format, thereby simplifying the modeling of temporal dependencies.

Our approach achieves comparable or improved performance over prior methods while using a simpler architecture. The scanline representation captures the temporal dynamics of the grinding process without requiring recurrent components, achieving a mean absolute error of 0.025 grams per gram of ground coffee compared to 0.027 for Deep Learning models using Recurrent Networks for temporal modeling and 0.064 for traditional regression baselines.

1. Introduction

Accurate estimation of granular material weight, such as ground coffee during the grinding process, is an important requirement. Conventional approaches rely on high-precision weighing scales to measure the cumulative mass of coffee. While effective, these systems are often expensive to manufacture and are sensitive to external factors such as vibrations and mechanical disturbances, which can introduce measurement errors. These limitations motivate the exploration of vision-based alternatives, where cameras are used to estimate weight.

Existing computer vision approaches for coffee weight estimation operate on video recordings of the grinding process. Early methods rely on traditional image processing techniques, such as pixel-based or frame-counting heuristics, to approximate the amount of material flow. More recent approaches adopt deep learning models [11], typically combining convolutional neural networks for spatial feature extraction with recurrent architectures to model temporal dependencies across frames [6]. Such CNN–RNN formulations have been widely used for video understanding tasks, where spatial and temporal features are jointly learned [1]. A common characteristic of these methods is that they process video data on a per-frame basis, extracting features from individual frames and subsequently aggregating temporal information to estimate the final output.

In this work, we explore an alternative formulation based on a scanline representation of the entire video sequence. Specifically, we transform a sequence of frames into a single structured image by stacking temporally ordered horizontal slices, with the goal of preserving both spatial content and temporal progression in a compact form. This idea is conceptually related to prior work that encodes video dynamics into a single image representation [2], but differs in its explicit preservation of temporal ordering through structured stacking. The proposed representation is designed to capture the dynamics of the grinding process while avoiding explicit sequence modeling.

We conduct a series of experiments in which the resulting scanline representation of a video is directly used as input to a Convolutional Neural Network (CNN) for weight estimation. To construct this representation, we explore multiple strategies for extracting a representative scanline, or scanline-like feature, from each frame, aiming to retain the most relevant visual information. These per-frame representations are then stacked in temporal order to form a single image encoding the entire sequence. The resulting

representation enables end-to-end learning using standard CNN architectures without the need for recurrent components such as RNNs. Using this approach, we achieve a mean absolute error per gram of ground coffee of 0.025 grams in weight estimation, demonstrating its effectiveness in capturing both spatial and temporal characteristics of the process.

Our approach can be viewed as an extension of prior deep learning methods that rely on frame-wise processing followed by temporal aggregation using recurrent or sequence-based models [6]. Instead of modeling time as an additional dimension through specialized architectures, we encode temporal information directly into the spatial structure of the representation. This enables the use of simpler models while retaining the ability to capture temporal patterns present in the video.

More broadly, this work contributes to the body of research on vision-based analysis of coffee-related processes, including tasks such as grind size estimation and weight prediction. Beyond this domain, the proposed representation is applicable to industrial settings where the weight of fine-grained particles must be estimated in continuous flow scenarios. In such cases, material flow evolves over time, and capturing this temporal progression is essential for accurate measurement. By embedding temporal dynamics into a compact image representation, our approach offers a practical and efficient alternative for modeling these processes.

2. Related Work

Computer vision techniques have been applied to a range of coffee-related tasks, including roast degree classification, grind size estimation, and bean quality assessment. Roast classification has been addressed using color and texture cues combined with deep convolutional networks [13, 19]. Grind size estimation has been studied using both classical image processing techniques and CNN-based approaches to analyze particle distributions [14, 15].

The first structured attempt to estimate cumulative coffee weight from video data was introduced in the DLCV project [11]. Their approach combined a ResNet-50 encoder [10] with a GRU-based recurrent decoder [5] to model temporal accumulation, demonstrating the importance of temporal reasoning for this task.

2.1. Vision-Based Physical Quantity Estimation

Estimating physical quantities from visual observations is an important direction in computer vision, with applications in volume estimation, material understanding, and physical reasoning from images [17, 18, 24]. These approaches exploit visual structure, motion patterns, and appearance cues to approximate physical properties without relying on mechanical sensors.

In the context of dynamic scenes, prior work has shown that motion and temporal evolution can provide strong cues for inferring physical behavior and material properties [8]. For granular materials in particular, the visual density and structure of particle flow can provide useful signals for estimating accumulated mass.

Our work extends this paradigm to espresso grinding, where the falling coffee stream encodes implicit information about cumulative weight.

2.2. Temporal Modelling in Video

Temporal reasoning in video tasks is commonly addressed using recurrent neural networks such as LSTMs and GRUs [5]. Architectures such as Long-Term Recurrent Convolutional Networks (LRCN) combine convolutional feature extraction with recurrent sequence modeling to capture spatiotemporal dependencies [6]. This paradigm has also been adopted in prior work on coffee weight estimation [11].

Alternative approaches model temporal dynamics using 3D convolutional neural networks, which learn joint spatial-temporal filters directly [4, 21]. Other methods explore more efficient temporal modeling through sparse sampling or lightweight operations. Temporal Segment Networks (TSN) propose segment-based sampling to capture long-range temporal structure [23], while Temporal Shift Modules (TSM) enable temporal interaction within 2D CNN backbones [16].

More broadly, several works investigate aggregating temporal information into compact representations that can be processed by standard convolutional networks [2, 7, 9]. These approaches suggest that temporal dynamics can be captured without explicit sequential modeling.

Motivated by these directions, we explore a scanline-based representation that encodes temporal information spatially, enabling the use of standard 2D convolutional architectures.

2.3. Spatial Temporal Representations

Prior work has explored representing video sequences through compact image-like structures that encode temporal information. Early approaches such as visual rhythm and temporal slice representations construct spatio-temporal images by extracting scanlines or pixel profiles from individual frames and stacking them over time [3, 22]. Related methods in action recognition and video summarization have used temporal templates and dynamic images to encode motion information into single frames [2, 7].

More recent approaches extend this idea within deep learning frameworks by stacking per-frame features into 2D maps, enabling convolutional networks to process temporal information without explicit sequence modeling [20]. These representations demonstrate that temporal structure can be preserved through spatial organization.

Building on these ideas, our work applies a scanline-based representation to the problem of coffee weight estimation, where fine-grained temporal dynamics of particle flow are encoded into a structured image.

2.4. Temporal Subsampling Strategies

Temporal sampling strategies play an important role in video learning, particularly when full frame sequences contain significant redundancy. Several works demonstrate that sampling representative segments from videos can preserve temporal dynamics while improving efficiency [4, 23]. Random temporal cropping and segment-based sampling have also been shown to improve generalization by exposing models to diverse temporal contexts during training [12].

Beyond efficiency, temporal sampling is also used as a form of data augmentation, improving robustness to variations in sequence length and temporal alignment [21]. These approaches encourage models to rely on meaningful visual cues rather than fixed frame positions.

Inspired by these ideas, we explore multiple subsampling strategies, including random sampling, prefix-based sampling, and fixed cropping, to increase training diversity and analyze their effect on cumulative weight estimation.

2.5. Coffee Grinding Video Dataset

We build upon a dataset introduced by prior work [11], consisting of video recordings of the coffee grinding process. Each video is provided as a sequence of cropped frames, accompanied by cumulative weight labels for each frame, obtained using OCR-based measurements along with confidence scores. Additional metadata, including grind size and roast degree, is also provided for each sequence.

Dataset is recorded in a controlled environment using a single coffee grinder where the camera and lighting setup remained consistent across recordings. Coffee with three different roast degree and three values for grind size were used in the Dataset. Equal number of samples for each grind size and roast degree were selected in Dataset. Grind size significantly affects the structure of the particle stream, with coarser grinds forming larger clumps and finer grinds producing more dispersed particle distributions. Variations in roast degree further influence the appearance of the coffee when accumulated, contributing to diversity in the dataset. This dataset forms the basis for training and evaluating vision-based models for coffee weight estimation.

3. Dataset and Preprocessing

For the task of weight estimation from coffee grinding video samples, we utilize the dataset introduced in [11], consisting of video sequences corresponding to a total mass of 32 grams per recording. To obtain samples across different weight ranges, we construct multiple datasets using an overlapping sub-sampling strategy applied to the original video sequences.

Specifically, two primary subsets are derived: one covering a range of 4–10 grams with 25 sub-videos per sequence, and another covering 15–21 grams with 10 sub-videos per sequence. The first subset contains 3750 training and 1875 test samples, while the second consists of 1500 training and 750 test samples. In addition, we generate a third dataset spanning a broader range of 9–25 grams by applying a similar sub-sampling approach, resulting in 1500 training and 750 test samples. These subsets enable evaluation across both narrow and extended weight ranges, capturing different stages of the grinding process.

3.1. Scanline Representation from Subsamples

For each subsampled video sequence, we construct a scanline representation by transforming the temporal sequence of frames into a single structured image. Specifically, a fixed-height horizontal slice is extracted from each frame via vertical averaging, and these slices are stacked in temporal order to form a scanline image. This representation encodes temporal progression along the vertical axis while preserving spatial information relevant to particle distribution.

3.2. Subsampling Strategies

To further increase the diversity of training samples and improve generalization, we explore additional subsampling strategies applied to scanline representations. Given a scanline image corresponding to a full or partial sequence, smaller crops are extracted along the temporal axis, with each crop associated with the corresponding accumulated weight. We investigate the following strategies:

- **Random sampling:** Random temporal crops are extracted to increase variability in training samples.
- **Fixed cropping:** Crops are extracted from predefined temporal regions to maintain consistent alignment between visual content and target weight.

These strategies enable the model to learn from different temporal segments of the grinding process while effectively expanding the training dataset. Figure 3 (a) illustrates example frames from a coffee grinding sequence, while Figure 1 shows the corresponding scanline representations constructed from video samples.

4. Methodology

In this section, we describe the baseline architecture and the proposed scanline-based approach for estimating cumulative coffee weight from video data. We begin with a CNN–RNN baseline that models temporal dynamics explicitly, followed by our scanline representation that encodes temporal information spatially. We further introduce several design choices and experimental variations, including

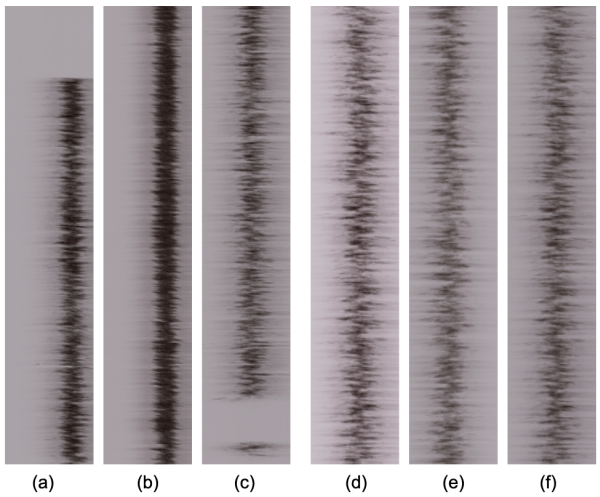

Figure 1. Scanline images for subsamples. (a)–(c) The three columns correspond to different portions of the pouring sequence, representing the start, middle, and end of the pour. Each image encodes temporal progression along the vertical axis, while darker regions indicate higher particle density, corresponding to heavier flow. (d)–(f) Scanline representations under different augmentation settings. (d): No flip, brightness increased by 17%, contrast increased by 11%, saturation increased by 4%. (e): Horizontal flip, brightness increased by 7%, contrast decreased by 14%, saturation increased by 4%. (f): No flip, brightness increased by 0.5%, contrast decreased by 7%, saturation increased by 4%.

subsampling strategies and quarter-split representations, to analyze how different forms of temporal aggregation affect performance.

4.1. Baseline Model: CNN-RNN Architecture

As a reference, we consider the baseline architecture proposed in prior work [11], which models temporal dynamics explicitly using a combination of convolutional and recurrent networks.

Each video frame is processed using a ResNet-50 encoder [10] to extract high-level visual features. To balance feature generalization and task-specific adaptation, all layers of the pre-trained network are frozen except for the final residual block. This allows the model to retain general visual representations while adapting to the domain-specific characteristics of coffee particle flow.

The resulting frame-level embeddings are then passed sequentially to a gated recurrent unit (GRU) [5], which acts as a temporal decoder. The GRU aggregates information across frames, capturing temporal dependencies in the grinding process and producing a final representation used for weight regression.

4.2. Scanline Approach

To avoid explicit sequence modeling, we propose a scanline-based representation that encodes temporal information spatially and enables efficient learning using standard convolutional networks.

For each frame, we extract a compact representation by applying vertical averaging across the frame height. This operation reduces the 2D spatial structure into a 1D horizontal profile while preserving the distribution of coffee particles across the width of the frame.

Formally, for a frame $I \in \mathbb{R}^{H\times W\times C}$, the scanline row $S \in \mathbb{R}^{1\times W\times C}$ is computed as:

$$S(x) = \frac{1}{H}\sum_{y=1}^{H} I(y, x) \tag{1}$$

The scanlines extracted from consecutive frames are stacked in temporal order to form a 2D image, where the vertical axis represents time. This representation transforms a variable-length video sequence into a structured image that preserves temporal ordering while enabling spatial processing.

The resulting scanline image is used as input to a ResNet-50 backbone [10], which extracts hierarchical features capturing both spatial patterns and temporally encoded dynamics. Unlike recurrent approaches, temporal dependencies are implicitly encoded through the spatial arrangement of the scanline representation.

The backbone output is processed using adaptive pooling to obtain a fixed-dimensional representation, followed by a regression head composed of fully connected layers. The model predicts a scalar value corresponding to the cumulative coffee weight, with a sigmoid activation applied to constrain the output within the target range.

Fig. 2 describes the working architecture using scanline images of the video samples.

4.3. Quarter-Split Representation

To capture finer temporal variations and retain more localized spatial information, we extend the scanline representation using a quarter-split formulation. The standard scanline approach performs vertical averaging over the entire frame, producing a single row per frame. While this provides a compact representation, it may discard useful spatial variations in the vertical direction that are relevant for modeling particle flow dynamics. To address this, we divide each frame into four horizontal regions and perform vertical averaging independently within each region. This results in four scanline rows per frame instead of one. These rows are then stacked sequentially, effectively increasing the temporal resolution of the scanline representation. In practice, this can be interpreted as extracting multiple temporal slices per frame, allowing the model to capture finer variations in the

flow of coffee particles. We choose four regions as a balance between preserving spatial detail and maintaining computational efficiency. Using too few splits reduces the benefit of localized information, while too many splits increases noise and redundancy without significant gains. We explore two implementations of this idea:

- **Pixel-level stacking:** The frame is divided into four horizontal regions, and vertical averaging is applied to each region independently. The resulting scanline rows are stacked along the temporal axis, forming a denser scanline image.
- **Feature-level stacking:** Instead of aggregating at the pixel level, we perform aggregation at a higher representation level to better preserve temporal structure.
  Let F denote the total number of frames in a sequence. Since each frame contributes four scanline rows, the initial stacked representation has the shape $T \in \mathbb{R}^{3\times 4F\times W}$, where 3 represents the RGB channels and W is the crop width.
  Rather than stacking these rows directly along the temporal axis, we partition the tensor into four distinct quarter-representations corresponding to the four spatial regions. Each slice takes the shape $S_i \in \mathbb{R}^{3\times F\times W}$ for $i \in \{1, 2, 3, 4\}$.
  These representations are then concatenated along the channel dimension:

$$T_{\text{final}} = \text{Concat}(S_1, S_2, S_3, S_4, \text{dim} = 0) \tag{2}$$

  The resulting tensor $T_{\text{final}} \in \mathbb{R}^{12\times F\times W}$ preserves temporal alignment across frames while separating spatial regions into distinct feature channels. This formulation allows the model to learn region-specific flow dynamics more effectively.

This quarter-split representation increases the amount of temporal information available to the model and enables it to learn more detailed flow dynamics across different spatial regions of the frame.

4.4. Learned Scanline Representation

Prior scanline approaches compress each video frame into a single row using a fixed column-wise mean, a non-learnable operation that assigns equal importance to all pixel locations and often dilutes sparse particle signals with background regions. To address this limitation, we propose a learned scanline representation that replaces this heuristic with a lightweight convolutional encoder.

Given an input frame $I_t \in \mathbb{R}^{H\times W\times 3}$ at time step t, we first extract a feature map using a shallow CNN. The encoder consists of a two-layer convolutional architecture with $3 \to 16$ channels, followed by Batch Normalization and ReLU activations.

To obtain a compact representation, we collapse the vertical spatial dimension using adaptive max pooling, producing a fixed-height feature row for each frame. Max pooling is preferred over average pooling as it preserves high-activation responses corresponding to sparse particle regions.

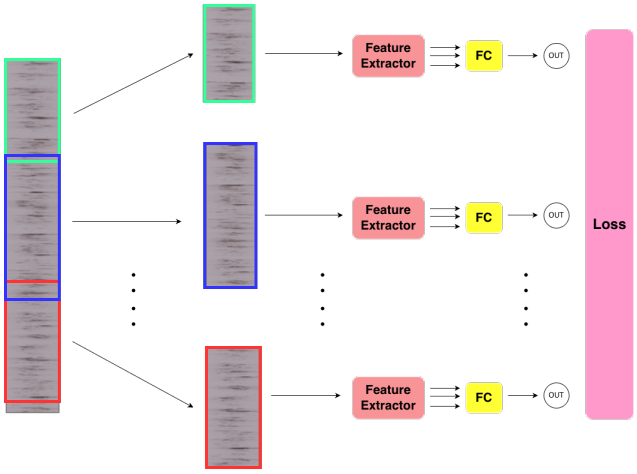

Figure 2. Overview of the scanline-based weight estimation architecture. A full scanline image of a video sample is subsampled(green, blue, red boxes), each representing a subsample window. Each crop is independently processed through a Feature Extractor followed by fully connected layers to predict the accumulated weight within that subsample window.

The per-frame representations are then stacked temporally to form a scanline tensor:

$$S = \text{Stack}(F_1, \ldots, F_N) \in \mathbb{R}^{16\times N\times W'}, \tag{3}$$

where F_t denotes the compact feature representation extracted from frame I_t, and N is the number of frames in the sequence.

This scanline tensor is used as input to a ResNet-50 backbone for regression. To accommodate the 16-channel input, the first convolutional layer is adapted by tiling pretrained weights across channels. Variable-length sequences are handled via symmetric padding, where padding is performed using pre-grind background frames to ensure consistent temporal scale without introducing spurious signal.

The network is trained to predict the cumulative coffee weight $\hat{y}$ using a regression head, where the ResNet-50 backbone is fine-tuned in the higher layers along with a fully connected output layer.

This formulation enables the model to capture both spatial and temporal patterns within a unified representation, while avoiding explicit sequence modeling.

4.5. Temporal Padding

The learned scanline approach requires passing variable-length subsample frame sequences through the CNN encoder, where each subsample corresponds to a pour of dif-

fering duration. To enable batch processing, these variable-length temporal tensors must be aligned to a consistent length. We pad each subsample to the maximum sequence length within its batch using a mean background frame, computed by averaging all cropped video frames captured before the grinding process begins. This provides a realistic and consistent representation of the empty grinder background, ensuring that padding frames remain in-distribution with respect to the camera setup and lighting conditions. Padding is applied symmetrically along the temporal axis, placing equal amounts of padding before and after the valid frames so that the physical pour remains centered within the input tensor regardless of subsample duration.

The observed degradation in model performance when applying the learned scanline approach to the 15–21g subsample dataset motivates a systematic investigation into the effect of padding on prediction accuracy. To this end, we conduct controlled injection experiments in which fixed proportions of padding, specifically 5% and 10% of the total sequence length, are introduced into samples. This allows us to quantify how increasing amounts of non-informative frame content affect regression performance. Motivated by these findings, we evaluate the learned scanline approach under a fixed-crop setting that requires no padding. As described in Section 3.2, window-based sampling is applied directly over the full video sequence, extracting fixed-length segments of 600 frames from the early, middle, and late stages of each pour without restricting the weight range. This setting eliminates the need for padding and naturally spans a broader weight range of 9–25g. It therefore provides a clean upper bound on the performance of the learned scanline approach, without the confounding effects introduced by padding.

4.6. Data Augmentation

To improve model generalization and robustness, we explore a set of data augmentation strategies applied at the frame level prior to constructing scanline representations. Since the proposed method relies on aggregating per-frame information into a structured image, it is important that augmentations preserve temporal consistency within each video subsample. Therefore, all transformations are applied consistently across frames of a given subsample. We consider geometric, photometric, and edge-enhancing augmentations, and analyze their impact on weight estimation performance.

Geometric Augmentation We apply horizontal flipping as a geometric augmentation, where each video subsample is mirrored along the horizontal axis. This augmentation aims to improve invariance to spatial orientation and camera viewpoint, reducing the model's reliance on a fixed directional flow of particles or specific scene layout. Fig. 1 (e) illustrates the scanline representation constructed from horizontally flipped frames reflects the mirrored particle flow while preserving the underlying temporal structure.

Photometric Augmentation To simulate variability in imaging conditions, we apply photometric augmentations by adjusting brightness, contrast, and saturation. The transformation parameters are sampled once per subsample and applied uniformly across all frames to maintain consistent lighting conditions. These augmentations target robustness to changes in illumination, camera exposure, and sensor characteristics, encouraging the model to learn features invariant to such variations. As shown in Fig. 1, the three scanline images (d), (e), (f) correspond to different brightness, contrast, and saturation settings sampled within these ranges, resulting in noticeable variations in intensity and particle appearance.

Edge-enhancing Augmentation We investigate edge-enhanced representations by applying the Sobel operator(add reference) to individual frames before scanline construction. This transformation emphasizes intensity gradients and highlights structural boundaries of coffee particles. The motivation is to encourage the model to focus on shape and motion cues rather than raw intensity values, thereby improving robustness to variations in texture and illumination. However, this process also alters the visual characteristics of the particles by reducing their apparent size and smoothing fine-grained density patterns, which may remove subtle cues relevant for accurate weight estimation. Fig. 3 depicts example frames with the Sobel operator applied demonstrate the resulting edge-enhanced representations used for scanline construction.

5. Results and Discussions

This section presents a comprehensive evaluation of the proposed methods for coffee weight estimation from grinding videos. The results are organized into three parts. First, we evaluate scanline image-based approaches, including the scanline and quarter-split methods, which operate on pre-constructed scanline representations, and compare them against prior CNN–RNN baseline [11]. Second, we present results for the learned scanline approach, which operates at the individual frame level, and analyze the impact of temporal padding on model performance. This analysis further motivates evaluation under a fixed-crop setting without padding.

Finally, we report results from data augmentation experiments within the scanline framework. Across all settings, we discuss the strengths and limitations of each approach in the context of practical coffee weight estimation.

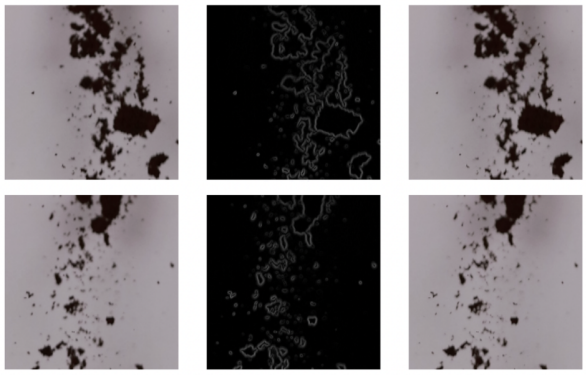

Figure 3. Effect of Sobel edge detection on sample frames. (a): raw sample frames from the coffee grinding dataset. Each frame (cropped to 384×384 pixels) captures the flow of ground coffee particles at different time steps. (b): edge maps obtained using the Sobel operator. (c): edge-enhanced frames highlighting structural boundaries while suppressing low-frequency intensity variations.

To assess prediction quality consistently across experiments and dataset settings, we report the Mean Absolute Error normalised by the ground truth weight of each sample, referred to as MAE per gram. This normalisation accounts for the variability in subsample lengths and weight ranges, producing a length-agnostic metric that quantifies the average prediction error relative to the actual amount of coffee dispensed. Alongside MAE per gram, we report the median and variance of this metric across test samples.

5.1. Scanline Image-Based Approaches

Tab. 1 reports results for the scanline image-based methods evaluated on the 15–21g subsample dataset. The quarter-split approach achieves the best performance with an MAE per gram of 0.025, marginally outperforming both the standard scanline method and the CNN-RNN baseline while using a comparably simple architecture.

The consistent performance across scanline variants suggests that the scanline representation encodes temporal dynamics as spatial structure, allowing the model to learn coffee particle flow patterns and preserve temporal ordering without requiring explicit sequential modelling components such as recurrent units. A practical limitation of these approaches is their dependence on offline scanline generation, which requires a preprocessing step to construct the scanline image representation for each video sample before training or inference can begin.

5.2. Learned Scanline and Padding Impact

The learned scanline approach applied to the 15–21g subsample dataset yields a MAE per gram of 0.0691, a notable degradation compared to the scanline image-based methods in Section 5.1. Given the variable subsample lengths in this dataset (300–845 frames), a mean of approximately 280 padding frames per sample is introduced during batch construction, as described in Section 4.5. We attribute this performance drop primarily to the high proportion of non-informative frames in the temporal scanline, which disrupts the feature distribution learned by the CNN encoder.

Method	MAE_G (↓)	Variance	Median
CNN-RNN [11]	0.027	**0.0005**	0.024
Standard Scanline	0.026	**0.0005**	0.022
Quarter Split	**0.025**	**0.0005**	**0.019**

Table 1. Comparison of scanline image-based methods and the CNN-RNN baseline on the 15–21g subsample dataset. Bold values indicate best performance.

Padding	MAE_G (↓)	Variance	Median
5%	**0.027**	**0.0006**	**0.022**
10%	0.029	0.0007	0.024

Table 2. Effect of injecting increasing proportions of mean background padding frames into the 15–21g subsample dataset on scanline-based regression performance. Bold values indicate best performance.

As detailed in Section 4.5, we conduct controlled padding injection experiments to validate this hypothesis. Tab. 2 shows that introducing 5% and 10% background padding into the 15–21g dataset leads to a progressive increase in MAE per gram from 0.027 to 0.029, suggesting towards non-informative frame content being direct contributor to the observed performance degradation rather than a limitation of the learned scanline approach itself. This observation is further supported by the in-between padding augmentation strategy used with the baseline architecture in [11], where short sequences of mean background frames are inserted at random temporal locations within the 15–21g benchmark, resulting in a significant degradation in CNN–RNN performance, with MAE per gram increasing to 0.082.

Following this finding, we evaluate the learned scanline approach under the fixed-crop setting described in Section 3.2, which eliminates padding entirely. As shown in Tab. 3, the model achieves an MAE per gram of 0.032 samples spanning 9.4–25.2g, comparable to the precision of prior specialised models while covering a significantly broader weight range within a single unified model. The removal of padding and the temporally aligned supervision provided by fixed cropping together enable the model to learn meaningful visual cues related to coffee particle accumulation flow.

A limitation of the fixed-crop approach is that the 600

Learned Scanline	MAE$_G$ (↓)	Variance	Median
Random (15-21g)	0.069	0.0024	0.059
Fixed Crop (9-25g)	**0.032**	**0.0006**	**0.026**

Table 3. Comparison of the learned scanline approach on the 15–21g random subsample dataset with variable-length padding and the fixed-crop setting spanning 9–25g. Bold values indicate best performance.

frame window constrains the model to sequences of fixed duration (10 seconds for 60 frames per second camera setting), which may limit generalisation to varying grind durations at inference time. The variable-length subsample setting, while more flexible, remains sensitive to the proportion of padding introduced.

5.3. Data Augmentation Study

Tab. 4 shows that introduction of augmentations leads to a decrease in performance compared to the model trained on unaugmented data. This outcome is expected, as the dataset is collected under a controlled setup, resulting in minimal variation between training and test samples. Nevertheless, these augmentations are important for improving robustness in more realistic deployment scenarios, where variations in camera setup, lighting conditions, and grinder configurations are likely to occur.

Photometric and geometric augmentations result in a mean absolute error (MAE) per gram of 0.052, indicating sensitivity to distributional shifts introduced by these simulated variations. Brightness and contrast are scaled by factors in the range [0.8, 1.2], while saturation is varied within [0.9, 1.1]. For Horizontal flipping we use a probability of 0.5 for each subsample. Changes in brightness, contrast, saturation, and spatial orientation alter the visual characteristics of the particle flow, which affects the consistency of the scanline representation and degrades performance.

The Sobel-based augmentation performs relatively better, achieving an MAE per gram of 0.034, but still underperforms the baseline scanline model (0.026). While edge enhancement improves the visibility of structural boundaries, it also reduces the apparent size and density of coffee particles, potentially removing fine-grained visual cues essential for accurate estimation. As shown in Fig. 3, the Sobel operator emphasizes particle boundaries while suppressing low-frequency intensity variations, which impacts the quality of features available for prediction.

The applied augmentations do not improve performance, reflecting the limited variability in the dataset due to its controlled acquisition setup. However, they remain important for improving generalization to real-world conditions, including variations in lighting, camera settings, and scene configuration.

Augmentation	MAE$_G$ (↓))	Variance	Median
None	**0.026**	**0.0005**	**0.024**
Edge Enhancement	0.034	0.0008	0.029
H. Flipping + Photometric	0.052	0.0018	0.044

Table 4. Quantitative impact of data augmentation strategies on the standard scanline architecture (Section 4.2) evaluated on the 15–21g subsample dataset. Results compare scanlines generated without augmentation, with edge enhancement via Sobel filtering, and with combined geometric and photometric augmentation. Bold values indicate best performance.

6. Conclusion and Future Work

In this work, we addressed the problem of cumulative coffee weight estimation from visual data by introducing a scanline-based representation that encodes temporal dynamics into a compact spatial format. By transforming a sequential modelling problem into a standard image regression task, our approach eliminates the need for recurrent architectures and reduces overall model complexity. We further proposed a learned scanline method, wherein individual frames are first processed by a CNN encoder and the resulting feature rows are stacked temporally, enabling the model to learn spatially informative compressions rather than relying on fixed averaging.

Our scanline-based methods achieve an MAE per gram of 0.025, surpassing CNN-RNN baselines while maintaining a simpler architecture. Through systematic evaluation of subsampling strategies, we demonstrate that temporally aligned fixed cropping allows the learned scanline model to generalize across a wider weight range within a single unified framework. Controlled experiments also reveal that padding introduced during variable-length batch construction consistently degrades performance across architectures. These results highlight the simplicity and effectiveness of the scanline representation, making it particularly well-suited for non-contact industrial material flow estimation.

A notable limitation of the learned scanline approach is its sensitivity to padding when subsamples vary substantially in length. While fixed cropping mitigates this issue, it imposes a constraint on sequence duration at inference. Future directions of this work include developing sampling strategies that mitigate or adapt to padding while supporting arbitrary video lengths, with temporal adaptive pooling over encoded frame features can be promising. In addition, applying scanline-based temporal encoding to other granular material estimation tasks and evaluating its robustness across different camera setups and environmental conditions will further demonstrate the generality and applicability of the approach.

References

[1] Khaled Alomar, Halil Ibrahim Aysel, and Xiaohao Cai. CNNs, RNNs and transformers in human action recognition: A survey and a hybrid model. *Artificial Intelligence Review*, 2025. 1

[2] Hakan Bilen, Basura Fernando, Efstratios Gavves, and Andrea Vedaldi. Dynamic image networks for action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016. 1, 2

[3] Aaron F. Bobick and James W. Davis. The recognition of human movement using temporal templates. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2001. 2

[4] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 2, 3

[5] Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder–decoder for statistical machine translation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 2014. 2, 4

[6] Jeff Donahue, Lisa Anne Hendricks, Sergio Guadarrama, Marcus Rohrbach, Subhashini Venugopalan, Kate Saenko, and Trevor Darrell. Long-term recurrent convolutional networks for visual recognition and description. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015. 1, 2

[7] Basura Fernando, Efstratios Gavves, Jose M. Oramas, Amir Ghodrati, and Tinne Tuytelaars. Modeling video evolution for action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015. 2

[8] Katerina Fragkiadaki, Pulkit Agrawal, Sergey Levine, and Jitendra Malik. Learning visual predictive models of physics for playing billiards. In *Proceedings of the International Conference on Learning Representations*, 2016. 2

[9] Rohit Girdhar, Deva Ramanan, Abhinav Gupta, Josef Sivic, and Bryan Russell. Actionvlad: Learning spatio-temporal aggregation for action classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 2

[10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016. 2, 4

[11] L. Kammeyer, Frederick Wichert, and Jonas Milkovits. Deep learning-based coffee grinding weight estimation. DLCV Project Report, 2025. 1, 2, 3, 4, 6, 7

[12] Andrej Karpathy, George Toderici, Sanketh Shetty, Thomas Leung, Rahul Sukthankar, and Li Fei-Fei. Large-scale video classification with convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014. 3

[13] D. S. Leme, B. H. G. Barbosa, et al. Recognition of coffee roasting degree using a computer vision system. *Computers and Electronics in Agriculture*, 2019. 2

[14] Franky Leonard and Habibullah Akbar. Coffee grind size detection using convolutional neural network architecture. *Journal of Applied Science, Engineering, Technology, and Education*, 2022. 2

[15] Parkpoom Lertsawatwicha and Thitirat Siriborvornratanakul. Measuring particle size distribution of ground coffee using computer vision. *International Journal of Information Technology*, 2023. 2

[16] Ji Lin, Chuang Gan, and Song Han. Temporal shift module for efficient video understanding. In *Proceedings of the IEEE International Conference on Computer Vision*, 2019. 2

[17] Austin Meyers, Nick Johnston, Vivek Rathod, Anelia Korattikara, Alex Gorban, Nathan Silberman, Sergio Guadarrama, and Kevin P. Murphy. Im2calories: Towards an automated mobile vision food diary. In *Proceedings of the IEEE International Conference on Computer Vision*, 2015. 2

[18] Roozbeh Mottaghi, Connor Schenck, Dieter Fox, and Ali Farhadi. See the glass half full: Reasoning about liquid containers, their volume and content. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017. 2

[19] S. Ontoum et al. Coffee roast intelligence. arXiv preprint arXiv:2206.01841, 2022. 2

[20] Xiaolin Song, Cuiling Lan, Wenjun Zeng, Junliang Xing, Xiaoyan Sun, and Jingyu Yang. Temporal–spatial mapping for action recognition. *IEEE Transactions on Circuits and Systems for Video Technology*, 2020. 2

[21] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3d convolutional networks. In *Proceedings of the IEEE International Conference on Computer Vision*, 2015. 2, 3

[22] Ba Tu Truong and Svetha Venkatesh. Video abstraction: A systematic review and classification. *ACM Transactions on Multimedia Computing, Communications, and Applications*, 2007. 2

[23] Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, and Luc Van Gool. Temporal segment networks: Towards good practices for deep action recognition. In *Proceedings of the European Conference on Computer Vision*, 2016. 2, 3

[24] Jiajun Wu, Ilker Yildirim, Joseph J. Lim, Bill Freeman, and Joshua Tenenbaum. Galileo: Perceiving physical object properties by integrating a physics engine with deep learning. In *Advances in Neural Information Processing Systems*, 2015. 2