Doppio: A Dataset for Contactless Weight Estimation of Falling Particles

Simon Kiefhaber*,1 (✉), Jan-Martin O. Steitz*,1, Julia Grabinski[1], Christoph Reich[1,2,3,4], Paul Wagner[1], Max Zimmermann[1], Simone Schaub-Meyer[1,3,5], and Stefan Roth[1,3,5]

[1] TU Darmstadt [2] TU Munich [3] ELIZA [4] MCML [5] hessian.AI * equal contribution
{name.surname}@visinf.tu-darmstadt.de
https://visinf.github.io/doppio

Abstract. Measuring the mass of powder, including falling particles, is a common task in industrial applications. While scales are effective for static measurements, many applications require contactless sensing, where existing solutions are often costly, application-specific, and technically complex. In this work, we investigate computer vision as a practical alternative for contactless mass estimation. As an accessible real-world case study, we focus on coffee grinding and introduce *Doppio*, a novel video dataset capturing videos of falling ground coffee, paired with precise, per-frame ground-truth weight measurements. To demonstrate contactless measuring, we evaluate deep learning-based approaches ranging from purely spatial feed-forward networks to recurrent spatio-temporal models. These models are analyzed with respect to their predictive accuracy and computational trade-offs. We demonstrate that deep learning-based computer vision models accurately estimate the cumulative weight of falling particles, establishing a solid foundation for future vision-based contactless measurement solutions.

Keywords: Weight Estimation · Video Dataset · Coffee Analysis

1 Introduction

The precise measurement of powder mass is a fundamental task common in numerous industry applications, including solid dosage in pharmaceutical manufacturing [1], additive manufacturing via laser metal deposition [2], and the food industry [30]. Mechanical vibrations often prohibit the application of scales or weight cells. Additionally, they only support batch processing, restricting material flow. For a continuous mode of manufacturing or contactless applications, other specialized measurement approaches are required. These often include highly specialized hardware such as microwave radar or X-ray [12,25], significantly increasing engineering effort, technical complexity, and total production cost. A simple, cost-efficient, and general contactless measuring solution remains lacking.

Therefore, there is a significant incentive to move toward off-the-shelf contactless measurement solutions. Computer vision offers a compelling alternative.

Optical sensors are less prone to mechanical vibration and universally available as low-cost, high-resolution video cameras. Additionally, current computer vision approaches demonstrate effective video understanding [38,54]. Video-based weight estimation using computer vision can aid manufacturers reduce costs and technical complexity while increasing system durability.

As access to industry-scale systems that use powder flow is limited, we turn to coffee bean grinding as a *case study*. The grinding of coffee beans provides a relevant setting, as industry applications frequently rely on the dosing of fine-grained materials. Taking a closer look, during coffee grinding, coffee particles agglomerate mainly due to the triboelectric effect [41], but also because of capillary liquid bridges from the lipid fraction released from the coffee beans [20,52], and powder caking due to compaction in the grinder [8]. As a result, depending on agglomerate size, we observe different speeds and occlusions of smaller clumps and particles, making the weight estimation of falling ground coffee a challenging problem and, therefore, an adequate substitute model.

To the best of our knowledge, no published work has explored the use of computer vision for dynamic and contactless weight estimation of falling agglomerated particles. Our core contribution is to present and release the first specialized dataset for vision-based weight estimation of ground coffee: The *Doppio* dataset comprises videos capturing a diverse range of roast types as they fall from a grinder at multiple grind settings, providing a benchmark to facilitate future research in this domain. Along with the dataset, we present a range of models, from a simple time-based model to recurrent deep neural networks, to showcase that weight information can be extracted from the continuous video signal.

2 Related Work

Vision-Based Contactless Weight Estimation. To bypass the high cost and mechanical limitations of physical scales, estimation of mass through computer vision finds application in broader agricultural and industrial contexts [37,39,42]. In livestock monitoring, various approaches for weight estimation have been introduced, demonstrating a shift toward contactless sensors to reduce operational complexity [9]. In the context of vegetable and food processing, geometric feature mapping alongside multi-form shape properties has been used to estimate weights [27,43]. Crucially, when dealing with materials in motion, recent work has explored tracking frameworks; for instance, deep learning-based object detection has been used to estimate the cumulative mass flow of irregular crops moving rapidly along a harvester conveyor belt [26]. More broadly, deep learning-based end-to-end image-to-mass frameworks have successfully modeled the complex relationship between an item's visual geometry, volume, and hidden material densities to predict physical weight [6,53] or calories [19,28,40,45].

Mass Flow Estimation of Particles. In the domain of mass flow estimation for particles in industrial settings, it is common to employ measurement methods that use modalities other than RGB imagery [47]. Optical sensors, using

photoreceptors and lasers, have been employed to estimate mass flow rates by measuring the length of falling particle clusters [16]. For pharmaceutical tablet manufacturing, both capacitance-based sensing [24] and X-ray sensors [12] have demonstrated real-time, in-line mass flow rate monitoring in a non-invasive manner. Microwave Doppler radar has been applied to solid-flow measurements, with falling beans serving as a model [25]. Using an image-based analysis, Gao *et al.* [13] recover the particle size distribution from a laser-illuminated stream of pneumatically conveyed particles using a high-speed charge-coupled device camera and contour-based image processing. While these approaches demonstrate that reliable mass flow estimation and particle characterization are achievable in industrial pipelines, they generally rely on a pneumatically controlled mass flow or dedicated, specialized hardware (*e.g.*, X-ray, solid-state laser, or microwave Doppler radar).

Deep-Learning-Based Coffee Analysis. Translating visual volume into a precise weight metric requires accounting for particle size and material density. Due to limited access to industrial particle-flow systems, we utilize coffee grinding as an accessible case study. In the context of coffee processing, research has focused on analyzing the following granular variations. For example, coffee granularity classification has been approached using AlexNet [29,34].Exploring the use of simple consumer hardware [35] demonstrated that using traditional computer vision techniques, such as Canny edge detection [3] and connected components labeling [7], allows for measuring fine particles (200-1600 µm) using mobile-phone cameras. Following this approach, shape analysis was used on ground coffee to determine the distribution of particle sizes [44]. Further work in the coffee industry focused on classifying raw materials [4] and on quality control [22,36] during the roasting process. Several studies have leveraged deep learning architectures to categorize bean characteristics. For instance, a classification dataset comprising 8 k images of unroasted coffee beans was proposed to identify defects and bean varieties [11]. Regarding the roasting process, standard deep learning-based vision architectures such as LeNet [32], ResNet [18], DenseNet [23], and MobileNet-V2 [48] were trained and benchmarked on individual beans at various roast levels [17]. Similar classifications on images containing bulk quantities of roasted beans were also performed using neural networks [33]. While these works establish the effectiveness of using deep learning-based vision models for identifying coffee-specific features, they remain limited to static, non-temporal analysis.

While these studies focus on analyzing the coffee, they do not quantify how much is being ground in a dynamic setting. In this work, we extend these concepts by moving beyond static, categorized, or volumetric analysis toward spatiotemporal understanding for efficient, real-time contactless weight estimation of falling particles.

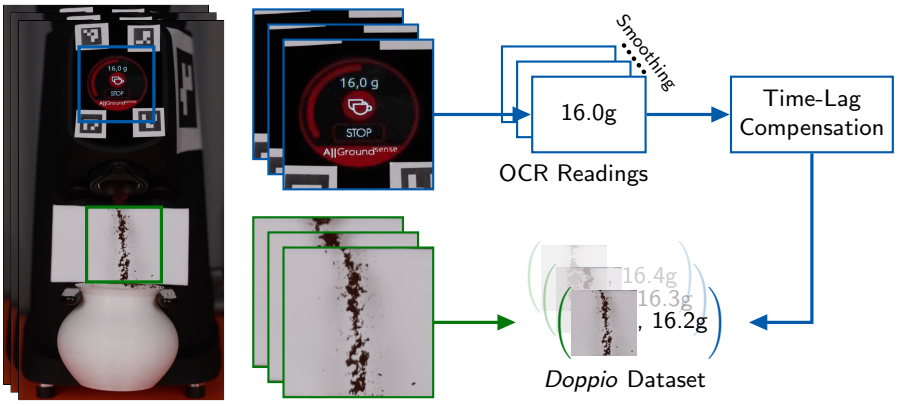

Fig. 1. Data acquisition pipeline for a single sequence of the *Doppio* dataset. Crops of the display and coffee are taken relative to the ArUco markers. The display crop is then processed by an optical character recognition (OCR) approach, and a time-lag compensation is applied. Finally, we construct a sequence of image-weight pairs.

3 The *Doppio* Dataset

Due to the lack of publicly available datasets containing videos of falling particles with per-frame weight annotations, we record a dedicated dataset—*Doppio*. We will release our *Doppio* dataset, evaluation metrics, and data loader with the publication of this paper. We use a coffee grinder to produce falling coffee particles and capture videos of the falling coffee. In these videos, we also capture the machine's display showing the cumulative weight of the coffee determined by an integrated scale.

3.1 Data Acquisition

For our recordings, we use a Fiorenzato AllGround Sense coffee grinder. This grinder offers discrete grind size adjustments and an integrated scale, well-suited for our data acquisition. The scale provides the real-time weight measurements, which we record to obtain ground-truth weight measurements. To ensure sufficient variability, we record sequences spanning 25 different grind sizes and 3 different types of coffee beans. Each video in the dataset shows a continuous grinding process that yields about 32 g of ground coffee. Figure 1 *(left)* provides an overview of our data acquisition setup.

To ensure a consistent spatial arrangement of our capturing setup, we attach ArUco markers [14] to all critical components in the scene. This includes the floor, the main camera, the grinder, the lighting, and the tripods. For robust pose estimation and tracking, each object is equipped with at least three markers. A Logitech C920 webcam is used to detect these markers and maintain spatial calibration between recordings.

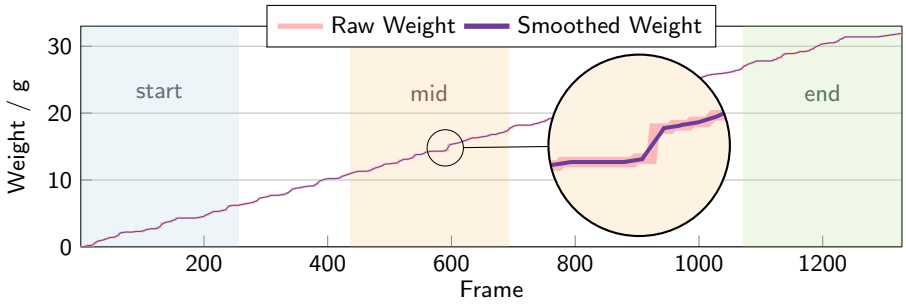

Fig. 2. Example weight sequence. Visualization of the raw and smoothed weights of one exemplary sequence of our *Doppio* dataset. Additionally, the *start* ■, *mid* ■, and *end* ■ sections are highlighted.

To record the main sequence of falling particles, it is important to capture the fast-moving coffee particles without motion blur, so they remain clearly visible. Therefore, we employ a Canon EOS R5 camera, recording at 60 fps at a 4 K resolution, with a fast shutter speed of 1/2000 s, an aperture of f/3.2, and an ISO value of 1000. In Fig. 1, we show an example frame captured by the camera with these settings.

3.2 Postprocessing and Annotation

We extract a 512×512 px crop of the falling coffee from each frame. To determine the cumulative weight, we extract a second crop that contains the machine's display (*cf*. Fig. 1 *(middle)*). Then, we apply an optical character recognition (OCR) pipeline based on Tesseract OCR [51] to read the display's content. Since the weight is unreadable in some frames due to the display refreshing its content while the frame is captured, we filter out OCR misdetections and fill missing values. We identify misdetections by assuming a monotonic growth in weight and by limiting the maximum increase between consecutive frames. The missing values are then linearly interpolated from their closest valid neighbors.

We also compensate for the time lag between when the ground coffee appears in the camera frame and when the weight is measured (*cf*. Fig. 1 *(right)*). This offset is caused by the time it takes for the coffee to fall onto the scale after passing through the cropped window. To that end, we estimate and apply a per-sequence time offset. This offset is estimated per sequence using the first frame in which ground coffee appears and the frame in which the scale detects an initial weight.

Since the integrated scale measurements show only a single digit after the decimal point, we apply a moving average window of seven frames to the raw measurements as a smoothing operator. This avoids ramp-like trajectories in our data. An illustrative example of this smoothing effect is presented in Fig. 2.

Table 1. Dataset statistics overview. We report statistics for training, validation, and test splits: mean weight increment and standard deviation per frame, minimum and maximum number of frames per sequence, and number of sequences per split.

Split	Mean (g/frame)	Std (g/frame)	Min frames	Max frames	Sequences
train	0.0304	0.0193	810	1406	131
val	0.0284	0.0196	979	1416	13
test	0.0303	0.0192	839	1480	75

3.3 Dataset Splits

We split our dataset into 131 training, 13 validation, and 75 test sequences. Further statistics of the splits are reported in Tab. 1. We ensure each combination of grind size and coffee bean type is present in the training and test set.

We further split the validation and test sequences into the following sub-sequences: *Start* and *end* contain the first and last 256 frames of each sequence, respectively. *Mid* contains 256 consecutive frames sampled from in-between the start and end sections (*cf*. Fig. 2). The *full* setting contains the entire sequence. These sub-sequences enable more detailed model evaluations, as the grinder's behavior varies throughout the grinding process. For example, at the start, slightly fewer grounds fall than in the middle. Towards the end, the machine's internal control loop stops or slows down the grinding process to reach the predefined target weight more accurately (*cf*. Fig. 2 *(right top)*). We include a video of the complete grinding process in our supplemental material to demonstrate this.

3.4 Evaluation Metrics

For our evaluation metrics, we define a doppio (a double shot of espresso, a common espresso serving) as 16 g of ground coffee. We measure the mean absolute error (MAE) per doppio. We term this metric $\mathrm{MAE}_{⏀}$, and we define it as

$$\mathrm{MAE}_{⏀} = \frac{16}{N} \sum_{i=1}^{N} \frac{|w_i(L_i) - \hat{w}_i(L_i)|}{w_i(L_i)}, \tag{1}$$

where N is the number of sample sequences, $w_i(j)$ is the measured ground truth weight at the j-th frame for the i-th sequence. $\hat{w}_i(j)$ denotes the estimated weight. L_i is the length of the i-th sequence, and therefore $w_i(L_i)$ refers to the last element of the sequence, so the MAE is only calculated after the entire sequence is processed.

Since our dataset contains per-frame weight annotations, we can measure the MAE at each time step and calculate the area between the predicted and ground-truth weight measurements. To enable comparison of variable-length sequences, we normalize this metric by N, the number of frames within a sequence. We define this metric as

$$\mathrm{MAE}_{□} = \frac{1}{N} \sum_{i=1}^{N} \frac{1}{L_i} \sum_{j=1}^{L_i} |w_i(j) - \hat{w}_i(j)|. \tag{2}$$

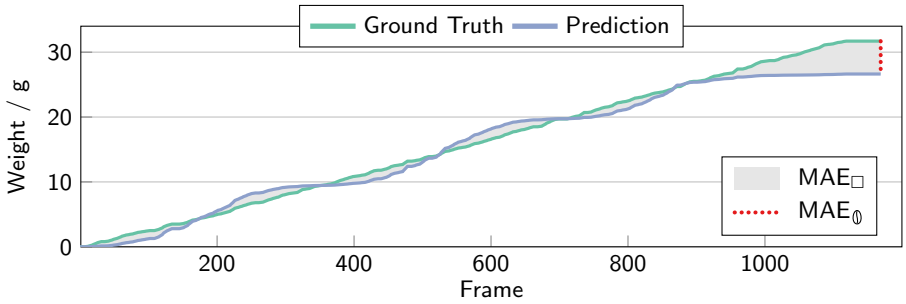

Fig. 3. Metrics visualization. $\text{MAE}_{\square}$ (gray shaded area) is calculated between the prediction and ground truth for each frame. $\text{MAE}_{◊}$ (red dashed line) is computed at the end of the sequence and measures the final discrepancy between the predicted and ground truth weight, while normalizing to the discrepancy per doppio (*i.e.*, 16 g).

We provide a visualization of how $\text{MAE}_{◊}$ and $\text{MAE}_{\square}$ relate to the sequence of frames in Fig. 3.

4 Methods for Contactless Weight Estimation

To systematically address the challenge of contactless weight estimation, we evaluate models of various complexity, ranging from simple linear regression to deep spatio-temporal architectures. First, we establish a naïve, time-based baseline to quantify the predictive boundaries achievable without any visual inputs, followed by more sophisticated computer vision models.

4.1 Time-Based Baseline

The naïve approach to estimating the weight of falling particles is a time-based model calibrated for each new combination of input material and particle size. We build this baseline model using linear regression for each possible combination of bean type and grind size, enabling us to assess whether more complex vision-based models are needed or whether the assumption of a constant fall rate is sufficient to solve this task.

4.2 Vision-Based Models

To predict the target weights across our video data, we process each frame individually to estimate the weight difference caused by the coffee visible at that time step. Per-frame differences are then accumulated using a cumulative sum to obtain the total weight of the processed sequence up to any given frame. For our task of estimating the cumulative weights of falling particles, we explore three vision-based approaches: a feed-forward approach that accumulates no temporal information, a recurrent architecture that can capture long sequences, and a temporal approach that considers multiple previous frames for its predictions.

Since the weight can only increase monotonically over time, we limit our models to output only positive weight estimates by using a softplus [10] activation function.

As inputs for all of our vision-based models, we use either a single RGB frame $I_t \in \mathbb{R}^{H \times W \times C}$ with $C = 3$ or a concatenation of the current frame and the pixel-wise difference between the current and previous frame, denoted as ΔI_t, doubling the channel dimension to $C = 6$.

Our feed-forward approach uses a standard feed-forward network (FFN) to directly predict the weight difference from only the input at the current time step, ignoring all previous frames. Standard FFNs lack temporal awareness, meaning that falling particles visible in multiple frames can skew the results. Thus, we also employ a recurrent architecture to track these particles accurately over time. We aim to capture long-term temporal dependencies by employing a gated recurrent unit (GRU) [5] coupled with a feed-forward backbone acting as a feature extractor. The extracted spatial features are passed to the GRU, which updates its hidden state and outputs the predicted weight differences. Since long-term temporal context may be unnecessary for this task, we also evaluate temporal convolutional networks (TCNs) [31] as an alternative to GRUs, using the same spatial features. In the TCNs, we apply causal convolutions along the time axis to also consider information from previous frames within a limited time window.

5 Experiments

5.1 Training Setup

For all experiments, we train our deep-learning-based models on sub-sampled sequences of 128 consecutive frames within each batch. Our training runs for 200 epochs using a one-cycle learning rate schedule with cosine annealing [50], with the first 10 % of iterations used for warmup.

To improve robustness and generalization, we apply random adjustments to the brightness and contrast of each sequence. Furthermore, we introduce a domain-specific temporal augmentation by randomly inserting frames containing no falling coffee into 10 % of the frames of each train sequence. This encourages the network to correctly predict a zero-weight delta when no particles are present.

As the learning objective, we employ a combined loss that supervises both the final accumulated weight and the intermediate per-frame differences, enforcing correct temporal dynamics. Our loss on the *last* accumulated weight is

$$\mathcal{L}_{\text{seq}} = d(w_i(L_i), \hat{w}_i(L_i)) \,, \tag{3}$$

where $d(x, y)$ measures the distance between x and y. $w_i(L_i)$ is the ground truth weight and $\hat{w}_i(L_i)$ the estimated weight, both at the end of the i-th sequence denoted by L_i. To enforce correct temporal dynamics, we compute a difference loss between the predicted weight deltas and the ground-truth *per frame* j:

$$\mathcal{L}_{\delta} = \frac{1}{L_i - 1} \sum_{j=2}^{L_i} d(w_i(j) - w_i(j-1), \hat{w}_i(j) - \hat{w}_i(j-1)) \,. \tag{4}$$

Table 2. Results on different input representations. We report results of our models with and without frame difference features ΔI_t on *Doppio* test, using $\text{MAE}_{⏀}$ and $\text{MAE}_{□}$ (both ↓) for the *full* sequences. We highlight the best results for each metric *per row* in orange.

	Backbone	Without ΔI_t		With ΔI_t	
		$\text{MAE}_{⏀}$	$\text{MAE}_{□}$	$\text{MAE}_{⏀}$	$\text{MAE}_{□}$
FFN	MobileNet-V3-S	0.28	0.39	0.19	0.25
	MobileNet-V3-L	0.29	0.37	0.20	0.26
	ResNet-18	0.24	0.33	0.26	0.30
	ResNet-34	0.25	0.33	0.21	0.24
GRU	MobileNet-V3-S	0.35	0.32	0.35	0.31
	MobileNet-V3-L	0.24	0.26	0.21	0.23
	ResNet-18	0.19	0.26	0.20	0.24
	ResNet-34	0.19	0.26	0.27	0.33
TCN	MobileNet-V3-S	0.31	0.37	0.24	0.31
	MobileNet-V3-L	0.19	0.23	0.29	0.33
	ResNet-18	0.18	0.23	0.18	0.24
	ResNet-34	0.18	0.23	0.17	0.23

Our final loss is a combination of $\mathcal{L}_{\text{seq}}$ and $\mathcal{L}_{\delta}$, weighted by a combination factor $\lambda \in \mathbb{R}^{+}$, *i.e.*, $\mathcal{L} = \mathcal{L}_{\text{seq}} + \lambda\mathcal{L}_{\delta}$. As the expected weights in our setting are numerically small, we use the smooth L1 [15] distance for $d(x, y)$ across all experiments. An evaluation of alternative distance functions is provided in the supplement.

Since all of our models described in Sec. 4.2 use a feed-forward network for feature extraction, we finetune ImageNet-pretrained [46] ResNets [18] and MobileNet-V3s [21] of different sizes on our proposed dataset. For our GRU architectures, we use a single GRU layer with a hidden dimension size of 256. In our TCNs, we use 4 causal convolution blocks with kernel size 5, hidden dimension 256, and a dilation that doubles in each block, starting at 1.

5.2 *Doppio* Evaluation Results

Design Choices. As described in Sec. 4.2, we propose two input modes for our approaches: one that uses only the current frame I_t, and another that uses the current frame and its pixel-wise difference with the previous frame, denoted as ΔI_t, stacked along the channel dimension. In Tab. 2, we analyze the impact of this design decision using different architectures in combination with various-sized backbones. For our feed-forward architectures, we find that using frame differences consistently improves the accuracy of all tested backbones except for ResNet-18 at very low computational cost, as only the first layer of each backbone needs to be modified to accommodate the six input channels. Given these results, we use the frame differences as additional inputs for all subsequent FFNs. The differences between the results of our GRU-based architectures in Tab. 2 are less pronounced, but given the large gap in accuracy for ResNets and

Table 3. Results on *Doppio*. We report results of our models on *Doppio* test, using $MAE_{◊}$ and $MAE_{□}$ (both ↓) for different sections and for the *full* sequences. Best results are highlighted for our FFNs, GRUs, and TCNs individually in orange and the overall best results over all of our models in red.

	Method	Full		Start		Mid		End	
		$MAE_{◊}$	$MAE_{□}$	$MAE_{◊}$	$MAE_{□}$	$MAE_{◊}$	$MAE_{□}$	$MAE_{◊}$	$MAE_{□}$
	Time baseline	0.54	0.57	0.98	0.25	0.75	0.20	2.95	0.68
FFN	MobileNet-V3-S	0.19	0.25	0.29	0.12	0.29	0.13	0.57	0.12
FFN	MobileNet-V3-L	0.20	0.26	0.29	0.12	0.28	0.13	0.57	0.11
FFN	ResNet-18	0.26	0.30	0.36	0.14	0.28	0.13	0.64	0.13
FFN	ResNet-34	0.21	0.24	0.32	0.13	0.23	0.12	0.63	0.13
GRU	MobileNet-V3-S	0.35	0.32	0.37	0.13	0.32	0.13	0.75	0.13
GRU	MobileNet-V3-L	0.24	0.26	0.31	0.12	0.28	0.13	0.59	0.12
GRU	ResNet-18	0.19	0.26	0.35	0.13	0.24	0.13	0.55	0.12
GRU	ResNet-34	0.19	0.26	0.31	0.12	0.26	0.13	0.56	0.12
TCN	MobileNet-V3-S	0.31	0.37	0.44	0.16	0.35	0.14	0.51	0.11
TCN	MobileNet-V3-L	0.19	0.23	0.27	0.12	0.24	0.12	0.47	0.11
TCN	ResNet-18	0.18	0.23	0.31	0.12	0.23	0.12	0.47	0.11
TCN	ResNet-34	0.18	0.23	0.30	0.12	0.22	0.12	0.46	0.11

the rather small gap for MobileNets, we chose to use the simpler inputs without frame differences in our subsequent GRU-based architectures. Analogously, for the TCN-based architectures, we observe a significant improvement in the accuracy of the MobileNet-V3-L backbone when the frame differences ΔI_t are excluded. Therefore, we decided not to use them in this setting.

We further investigate the choice of loss weighting factor λ for the different architecture-types and find that $\lambda = 10$ performs best for the feed-forward network, while $\lambda = 100$ is best for GRU- and TCN-based architectures. Therefore, we use these differing hyperparameters for all subsequent experiments. For completeness, we provide the full tables for these experiments in the supplement.

Main Results. Tab. 3 presents our main performance evaluation. All of our proposed vision-based architectures consistently outperform the *Time baseline* across all sub-sequences and metrics. When analyzing the purely spatial feed-forward networks (FFNs), the MobileNet-V3-S architecture achieves the highest overall accuracy. However, the performance margins are small, as the MobileNet-V3-L and ResNet-34 architectures achieve nearly identical accuracies.

When moving to temporal models, the trend shifts towards larger backbones, as all ResNet-based models outperform their MobileNet counterparts. In the case of GRUs, all MobileNet-based models are worse than in the FFN setting. This is likely due to the aggressive feature reduction, which, on the one hand, allows for high compute efficiency in MobileNets, but, on the other hand, limits the models' outputs to rather coarse features, lacking fine-grained details. For ResNets, we observe higher overall accuracies when used as backbones for GRUs and TCNs

Table 4. Leave-one-out validation. We train our TCN (w/ ResNet-34) only on training samples of certain types of coffee beans and report the overall $MAE_{\emptyset}$ (↓) on the *full* test sequences of *Doppio*. We also evaluate each type of bean individually and report the difference to the overall $MAE_{\emptyset}$ indicated in gray ■.

Training Split	All	A	B	C
{A, B, C}	0.18	0.18 (+0.00)	0.18 (+0.00)	0.18 (+0.00)
{A, B}	0.22	0.23 (+0.01)	0.20 (−0.02)	0.24 (+0.02)
{A, C}	0.38	0.31 (−0.07)	0.44 (+0.06)	0.41 (+0.03)
{B, C}	0.28	0.30 (+0.02)	0.27 (−0.01)	0.28 (+0.00)

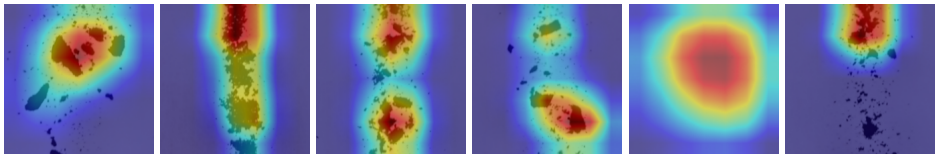

Fig. 4. Saliency visualization. Grad-CAM maps [49] of our TCN with a ResNet-34 backbone. Saliency is mostly located in regions with falling coffee grounds, if present. Color encoding: ■■■ (low → high saliency).

than when used as backbones for FFNs. In fact, our overall best model is a TCN using a ResNet-34 backbone, scoring a $MAE_{\emptyset}$ of 0.18 on the full test sequence.

To confirm that our models track only the visible coffee per frame, we analyze our TCN (w/ ResNet-34) using Grad-CAM [49]. Figure 4 shows qualitative Grad-CAM results. In general, the model only pays attention to falling coffee if present in the frame, and it only relies on cues from the background when no coffee is present. However, the last sample also shows that the network sometimes only pays attention to larger lumps as they fall, ignoring smaller ones.

Generalization. All experiments until this point evaluate exactly the same types of coffee beans in the training and test sets. This leaves open whether our proposed methods overfit only to the three types of beans we captured or *generalize* to new types. To gain insights into the generalization capabilities, we train our best-performing model, the TCN with a ResNet-34 backbone, on subsets of our training dataset containing only two types, and we evaluate them on the full test set in Tab. 4. We find that the overall accuracy of these models is lower than that of the model trained on the full training set, but this is most likely due to the reduced number of training samples. More interestingly, we find that the $MAE_{\emptyset}$ of the unseen bean type is only slightly higher than the $MAE_{\emptyset}$ of the types contained in the respective training subset. This demonstrates that our methods can generalize to unseen bean types. Another insight from this experiment is that models trained on bean type B are significantly more accurate than those without it. We observe that this type of coffee tends to produce fewer lumps than the other two, and these data seem beneficial for overall training.

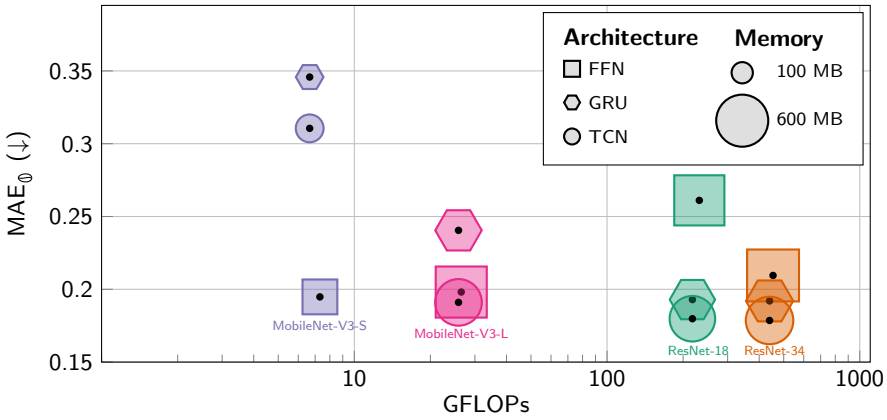

Fig. 5. Cost-accuracy tradeoffs of our proposed methods and backbones. We measure the memory and FLOPs for processing a 60-frame sequence (1 s of video) and report accuracy on the *full* test-set sequence.

Computational Cost *vs.* Accuracy. To evaluate trade-offs between computational cost and model accuracy, we measure the memory footprint and the number of required floating-point operations (FLOPs) for all proposed architectures on a 60-frame sequence (1 s of video material). The results of this evaluation are presented in Fig. 5. Our lightweight MobileNet-V3 backbone uses minimal computational power and memory; further, it delivers competitive accuracy with much larger models like ResNet-34 when used as FFN architecture. However, when we use these small backbones in our GRU- or TCN-based architecture, we see a large decrease in accuracy. This suggests that the highly compressed spatial features extracted by MobileNets lack the capacity needed to build meaningful recurrent states within the GRU and TCN.

In contrast, the ResNet-based FFNs perform worse than their GRU and TCN-based counterparts. Overall, the TCN with a ResNet backbone achieves the highest accuracy across all experiments, but comes at a drastically increased cost in required compute operations, compared to our more lightweight backbones.

6 Conclusion

In this paper, we present a case study on contactless weight estimation for falling particles using ground coffee. To enable this research, we captured and introduced *Doppio*, a dataset containing precise, per-frame ground-truth weight measurements of falling coffee grounds at different grind sizes and coffee bean types. We utilized our dataset to conduct a comprehensive evaluation of distinct architectures, including feed-forward networks, GRUs, and TCNs. We benchmarked a diverse range of backbone architectures, ranging from lightweight MobileNet-V3 variants to ResNets. Our analysis of computational costs highlights

the trade-offs between architectural families. We demonstrated that highly compressed models are well-suited for efficient feed-forward processing, but heavier convolutional networks are necessary to build meaningful temporal contexts in GRUs. Further, we demonstrate the generalization capabilities of our model to unseen types of beans. A promising direction for future work is the exploration of quantization-aware training and model quantizations to ensure these models can be efficiently deployed on low-cost, resource-constrained hardware. In conclusion, this work provides an easily accessible dataset and testbed for robust, real-time, vision-based mass estimation of falling particles.

Acknowledgments SK has been funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy – EXC-3057. J-MS has been funded by the State of Hesse through LOEWE emergenCITY (Grant no. LOEWE/1/12/519/03/05.001(0016)/72). JG has been funded by the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No. 866008). Christoph Reich is supported by the Konrad Zuse School of Excellence in Learning and Intelligent Systems (ELIZA) through the DAAD programme Konrad Zuse Schools of Excellence in Artificial Intelligence, sponsored by the German Federal Ministry of Education and Research. We also acknowledge the support of the European Laboratory for Learning and Intelligent Systems (ELLIS) and Munich Center for Machine Learning (MCML). SSM and PW have been funded by the DFG – project No. 529680848. Finally, we thank L. Kammeyer, J. Milkovits, and F. Wichert for their help with recording this dataset.

References

1. Blackshields, C.A., Crean, A.M.: Continuous powder feeding for pharmaceutical solid dosage form manufacture: A short review. Pharm. Dev. Technol. **23**(6), 554–560 (2018). https://doi.org/10.1080/10837450.2017.1339197
2. Breese, P.P., Hauser, T., Regulin, D., Seebauer, S., Rupprecht, C.: In situ measurement and closed-loop control for powder supply processes: Retrofittable solution in the context of laser metal deposition. Int. J. Adv. Manuf. Technol. **116**(3), 889–903 (2021). https://doi.org/10.1007/s00170-021-07438-z
3. Canny, J.: A computational approach to edge detection. IEEE TPAMI **8**(6), 679–698 (1986). https://doi.org/10.1109/TPAMI.1986.4767851
4. Chen, P., Jhong, S., Hsia, C.: Semi-supervised learning with attention-based CNN for classification of coffee beans defect. In: ICCE-TW. pp. 411–412 (2022). https://doi.org/10.1109/ICCE-Taiwan55306.2022.9869187
5. Cho, K., van Merrienboer, B., Gülçehre, Ç., Bahdanau, D., Bougares, F., Schwenk, H., Bengio, Y.: Learning phrase representations using RNN encoder-decoder for statistical machine translation. In: EMNLP. pp. 1724–1734 (2014). https://doi.org/10.3115/v1/D14-1179
6. Dehais, J., Anthimopoulos, M., Shevchik, S., Mougiakakou, S.: Two-view 3D reconstruction for food volume estimation. IEEE Trans. Multimed. **19**(5), 1090–1099 (2017). https://doi.org/10.1109/TMM.2016.2642792

7. Di Stefano, L., Bulgarelli, A.: A simple and efficient connected components labeling algorithm. In: ICIAP. pp. 322–327 (1999). https://doi.org/10.1109/ICIAP.1999.797615
8. Doğan, M., Aslan, D., Gürmeriç, V., Özgür, A., Saraç, M.G.: Powder caking and cohesion behaviours of coffee powders as affected by roasting and particle sizes: Principal component analyses (PCA) for flow and bioactive properties. Powder Technol. **344**, 222–232 (2019). https://doi.org/10.1016/j.powtec.2018.12.030
9. Dohmen, R., Catal, C., Liu, Q.: Computer vision-based weight estimation of livestock: A systematic literature review. N. Z. J. Agric. Res. **65**(2-3), 227–247 (2022). https://doi.org/10.1080/00288233.2021.1876107
10. Dugas, C., Bengio, Y., Bélisle, F., Nadeau, C., Garcia, R.: Incorporating second-order functional knowledge for better option pricing. In: NIPS. vol. 13, pp. 472–478 (2000), https://proceedings.neurips.cc/paper/2000/hash/44968aece94f667e4095002d140b5896-Abstract.html
11. Febriana, A., Muchtar, K., Dawood, R., Lin, C.Y.: USK-COFFEE Dataset: A multi-class green arabica coffee bean dataset for deep learning. In: CyberneticsCom. pp. 469–473 (2022). https://doi.org/10.1109/CyberneticsCom55287.2022.9865489
12. Ganesh, S., Troscinski, R., Schmall, N., Lim, J., Nagy, Z., Reklaitis, G.: Application of X-Ray sensors for in-line and noninvasive monitoring of mass flow rate in continuous tablet manufacturing. J. Pharm. Sci. **106**(12), 3591–3603 (2017). https://doi.org/10.1016/j.xphs.2017.08.019
13. Gao, L., Yan, Y., Lu, G.: Contour-based image segmentation for on-line size distribution measurement of pneumatically conveyed particles. In: I2MTC. pp. 1–5 (2011). https://doi.org/10.1109/IMTC.2011.5944318
14. Garrido-Jurado, S., Muñoz-Salinas, R., Madrid-Cuevas, F.J., Marín-Jiménez, M.J.: Automatic generation and detection of highly reliable fiducial markers under occlusion. Pattern Recognit. **47**(6), 2280–2292 (2014). https://doi.org/10.1016/J.PATCOG.2014.01.005
15. Girshick, R.B.: Fast R-CNN. In: ICCV. pp. 1440–1448 (2015). https://doi.org/10.1109/ICCV.2015.169
16. Grift, T.: Fundamental mass flow measurement of solid particles. Part. Sci. Technol. **21**(2), 177–193 (2003). https://doi.org/10.1080/02726350307492
17. Hassan, E.: Enhancing coffee bean classification: A comparative analysis of pretrained deep learning models. Neural Comput. Appl. **36**(16), 9023–9052 (2024). https://doi.org/10.1007/s00521-024-09623-z
18. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR. pp. 770–778 (2016). https://doi.org/10.1109/CVPR.2016.90
19. He, Y., Xu, C., Khanna, N., Boushey, C.J., Delp, E.J.: Food image analysis: Segmentation, identification and weight estimation. In: ICME. pp. 1–6 (2013). https://doi.org/10.1109/ICME.2013.6607548
20. Herminghaus, S.: Dynamics of wet granular matter. Adv. Phys. **54**(3), 221–261 (2005). https://doi.org/10.1080/00018730500167855
21. Howard, A., Pang, R., Adam, H., Le, Q.V., Sandler, M., Chen, B., Wang, W., Chen, L., Tan, M., Chu, G., Vasudevan, V., Zhu, Y.: Searching for MobileNetV3. In: ICCV. pp. 1314–1324 (2019). https://doi.org/10.1109/ICCV.2019.00140
22. Hsia, C.H., Lee, Y.H., Lai, C.F.: An explainable and lightweight deep convolutional neural network for quality detection of green coffee beans. Appl. Sci. **12**(21), 10966 (2022). https://doi.org/10.3390/app122110966

23. Huang, G., Liu, Z., Van Der Maaten, L., Weinberger, K.Q.: Densely connected convolutional networks. In: CVPR. pp. 2261–2269 (2017). https://doi.org/10.1109/CVPR.2017.243
24. Huang, Y.S., Medina-González, S., Straiton, B., Keller, J., Marashdeh, Q., Gonzalez, M., Nagy, Z., Reklaitis, G.V.: Real-time monitoring of powder mass flowrates for plant-wide control of a continuous direct compaction tablet manufacturing process. J. Pharm. Sci. **111**(1), 69–81 (2022). https://doi.org/10.1016/j.xphs.2021.06.005
25. Isa, M., Wu, Z.: Microwave Doppler radar sensor for solid flow measurements. In: EuMC. pp. 1508–1510 (2006). https://doi.org/10.1109/EUMC.2006.281364
26. Jang, S.H., Moon, S.P., Kim, Y.J., Lee, S.H.: Development of potato mass estimation system based on deep learning. Appl. Sci. **13**(4) (2023). https://doi.org/10.3390/app13042614
27. Kamiwaki, Y., Fukuda, S.: A machine learning-assisted three-dimensional image analysis for weight estimation of radish. Horticulturae **10**(2), 142 (2024). https://doi.org/10.3390/horticulturae10020142
28. Konstantakopoulos, F.S., Georga, E.I., Fotiadis, D.I.: A review of image-based food recognition and volume estimation artificial intelligence systems. IEEE Rev. Biomed. Eng. **17**, 136–152 (2024). https://doi.org/10.1109/RBME.2023.3283149
29. Krizhevsky, A., Sutskever, I., Hinton, G.E.: ImageNet classification with deep convolutional neural networks. Commun. ACM **60**(6), 84–90 (2017). https://doi.org/10.1145/3065386
30. Kruppa, F., Weiß, U., Oberdorfer, B., Wilke, B.: Increasing the dosing accuracy of a screw dosing device by inline measurement of the product density. Packag. Technol. Sci. **36**(3), 185–194 (2023). https://doi.org/10.1002/pts.2703
31. Lea, C., Flynn, M.D., Vidal, R., Reiter, A., Hager, G.D.: Temporal convolutional networks for action segmentation and detection. In: CVPR. pp. 1003–1012 (2017). https://doi.org/10.1109/CVPR.2017.113
32. Lecun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. Proc. IEEE **86**(11), 2278–2324 (1998). https://doi.org/10.1109/5.726791
33. Leme, D.S., da Silva, S.A., Barbosa, B.H.G., Borém, F.M., Pereira, R.G.F.A.: Recognition of coffee roasting degree using a computer vision system. Comput. Electron. Agr. **156**, 312–317 (2019). https://doi.org/https://doi.org/10.1016/j.compag.2018.11.029
34. Leonard, F., Akbar, H.: Coffee grind size detection by using convolutional neural network (CNN) architecture. J. Appl. Sci. Eng. Technol. Educ. **4**(1), 133–145 (2022). https://doi.org/10.35877/454RI.asci842
35. Lertsawatwicha, P., Siriborvornratanakul, T.: Measuring particle size distribution of ground coffee using computer vision. Int. J. Inf. Technol. **15**(6), 2961–2967 (2023). https://doi.org/10.1007/s41870-023-01364-x
36. Liang, C., Xu, Z., Zhou, J., Yang, C., Chen, J.: Automated detection of coffee bean defects using multi-deep learning models. In: APWCS. pp. 1–5 (2023). https://doi.org/10.1109/APWCS60142.2023.10234059
37. Liang, T., Yuan, Z.: Computer-vision-based non-contact paste concentration measurement. In: ICaMaL. pp. 1–9 (2024). https://doi.org/10.1109/ICaMaL62577.2024.10919773
38. Lin, J., Gan, C., Han, S.: TSM: Temporal shift module for efficient video understanding. In: ICCV. pp. 7083–7093 (2019). https://doi.org/10.1109/ICCV.2019.00718

39. Mathiassen, J.R., Misimi, E., Toldnes, B., Bondø, M., Østvik, S.O.: High-speed weight estimation of whole herring (Clupea harengus) using 3D machine vision. J. Food Sci. **76**(6), E458–E464 (2011). https://doi.org/10.1111/j.1750-3841.2011.02226.x
40. Meyers, A., Johnston, N., Rathod, V., Korattikara, A., Gorban, A., Silberman, N., Guadarrama, S., Papandreou, G., Huang, J., Murphy, K.P.: Im2Calories: Towards an automated mobile vision food diary. In: ICCV. pp. 1233–1241 (2015). https://doi.org/10.1109/ICCV.2015.146
41. Méndez Harper, J., McDonald, C.S., Rheingold, E.J., Wehn, L.C., Bumbaugh, R.E., Cope, E.J., Lindberg, L.E., Pham, J., Kim, Y.H., Dufek, J., Hendon, C.H.: Moisture-controlled triboelectrification during coffee grinding. Matter **7**(1), 266–283 (2024). https://doi.org/10.1016/j.matt.2023.11.005
42. Nyalala, I., Okinda, C., Nyalala, L., Makange, N., Chao, Q., Chao, L., Yousaf, K., Chen, K.: Tomato volume and mass estimation using computer vision and machine learning algorithms: Cherry tomato model. J. Food Eng. **263**, 288–298 (2019). https://doi.org/10.1016/j.jfoodeng.2019.07.012
43. Puri, M., Zhu, Z., Yu, Q., Divakaran, A., Sawhney, H.: Recognition and volume estimation of food intake using a mobile device. In: WACV. pp. 1–8 (2009). https://doi.org/10.1109/WACV.2009.5403087
44. Ren, Z., Zeng, J., Yang, Z., Tang, H., Wang, J., Jiang, L., Feng, W.: Digital image analysis for contact and shape recognition of coffee particles in grinding. Powder Technol. **440**, 119717 (2024). https://doi.org/10.1016/j.powtec.2024.119717
45. Ruede, R., Heusser, V., Frank, L., Roitberg, A., Haurilet, M., Stiefelhagen, R.: Multi-task learning for calorie prediction on a novel large-scale recipe dataset enriched with nutritional information. In: ICPR. pp. 4001–4008 (2021). https://doi.org/10.1109/ICPR48806.2021.9412839
46. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M.S., Berg, A.C., Fei-Fei, L.: ImageNet large scale visual recognition challenge. Int. J. Comput. Vis. **115**(3), 211–252 (2015). https://doi.org/10.1007/S11263-015-0816-Y
47. Samadi, M., Rostampour, V., Abdollahpour, S.: A review of solid particles mass flow rate measuring methods: Screening analytic hierarchy process for methods prioritization. J. Braz. Soc. Mech. Sci. Eng. **44**(8), 359 (2022). https://doi.org/10.1007/s40430-022-03663-z
48. Sandler, M., Howard, A.G., Zhu, M., Zhmoginov, A., Chen, L.: MobileNetV2: Inverted residuals and linear bottlenecks. In: CVPR. pp. 4510–4520 (2018). https://doi.org/10.1109/CVPR.2018.00474
49. Selvaraju, R.R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., Batra, D.: Grad-CAM: Visual explanations from deep networks via gradient-based localization. Int. J. Comput. Vis. **128**(2), 336–359 (2020). https://doi.org/10.1007/S11263-019-01228-7
50. Smith, L.N., Topin, N.: Super-convergence: Very fast training of neural networks using large learning rates. In: SPIE, Artif. Intell. Mach. Learn. Multi-Dom. Oper. Appl. vol. 11006, pp. 369–386 (2019). https://doi.org/10.1117/12.2520589
51. Smith, R.: An overview of the tesseract OCR engine. In: ICDAR. pp. 629–633 (2007). https://doi.org/10.1109/ICDAR.2007.4376991
52. Speer, K., Kölling-Speer, I.: The lipid fraction of the coffee bean. Braz. J. Plant Physiol. **18**(1), 201–216 (2006). https://doi.org/10.1590/S1677-04202006000100014

53. Standley, T., Sener, O., Chen, D., Savarese, S.: image2mass: Estimating the mass of an object from its image. In: CoRL. vol. 78, pp. 324–333 (2017), https://proceedings.mlr.press/v78/standley17a.html
54. Tang, Y., Bi, J., Xu, S., Song, L., Liang, S., Wang, T., Zhang, D., An, J., Lin, J., Zhu, R., et al.: Video understanding with large language models: A survey. IEEE Trans. Circuits Syst. Video Technol. **36**(2), 1355–1376 (2026). https://doi.org/10.1109/TCSVT.2025.3566695