PAMUKKALE ÜNİVERSİTESİ
Mühendislik Bilimleri Dergisi

Araştırma makalesi | Research paper

PAMUKKALE UNIVERSITY
Journal of Engineering Sciences

Cilt | Vol: XX • Sayı | No: X • ss | pp: XXXX-XXXX • Yıl | Year: 20XX

Contactless hemoglobin estimation from facial video using edge computing and sliding-window analysis

Ufuk Bal [1], Abdulbaki Akgün [2], Sahra Nur Pakel [2], Hüseyin Sezerol [2], Ahmet Çağdaş Seçkin [2], Alkan Bal [3]

ARTICLE INFO

Dates:
Received: 02.02.2026
Accepted: 01.04.2026

Doi:
10.65206/pajes.1880275

Corresponding author:
Ahmet Çağdaş Seçkin
(seckin.ac@gmail.com)

Author addresses:
[1] Department of Electrical and Electronics Engineering, Faculty of Engineering and Natural Sciences, Osmaniye Korkut Ata University, Osmaniye, 80000, Türkiye
(ufukbal@osmaniye.edu.tr)

[2] Department of Computer Engineering, Engineering Faculty, Aydın Adnan Menderes University, Aydın, 09010 Türkiye
(bakiakgun99@gmail.com; sahranurpakelx@gmail.com; h.sezerol60@gmail.com; seckin.ac@gmail.com)

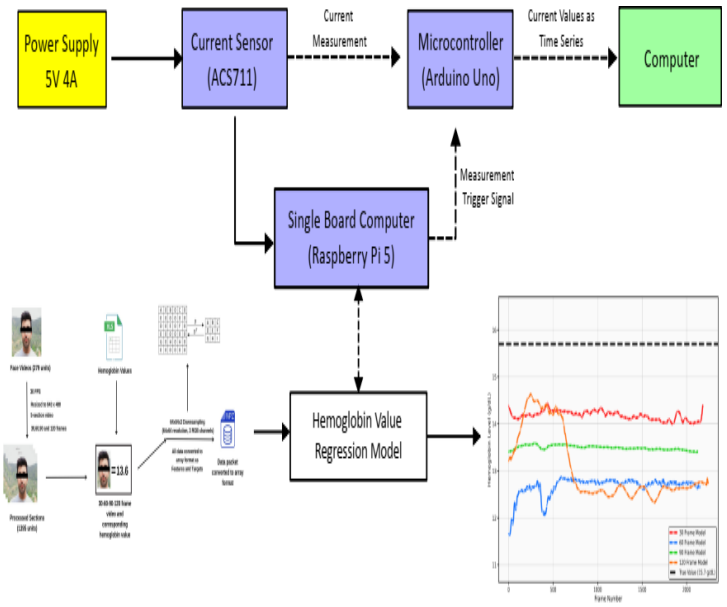
[3] Department of Pediatrics, Faculty of Medicine, Manisa Celal Bayar University, Manisa, 45140, Türkiye
(alkan.bal@cbu.edu.tr)

ABSTRACT

Context—Hemoglobin is a key biomarker for anemia screening and clinical decision support. Yet, standard measurement is invasive and can be difficult to deploy for rapid, repeatable assessment in field and disaster scenarios. This motivates contactless estimation approaches that can run reliably on resource-constrained edge devices.

Objective—This study investigates contactless peripheral hemoglobin (SpHb) estimation from facial videos with an explicit edge-computing viewpoint. The central aim is to quantify how sliding-window length acts as a system-level design parameter that jointly governs (i) prediction accuracy, (ii) temporal stability of sequential outputs, and (iii) power/energy cost on an embedded platform.

Method—The dataset contains 279 facial videos, each paired with a single SpHb label. Each video is divided into five segments, and training samples are generated using sliding-window configurations of 30, 60, 90, and 120 frames. Evaluation follows a subject-independent split to prevent data leakage. Algorithmically, 3D CNN baselines are compared with hybrid spatiotemporal architectures (CNN–VAN–Transformer). Performance is assessed using regression metrics (RMSE/MAE) and complemented by time-series inspection on representative videos to discuss output stability under different window lengths.



Results—On 60-frame segments, the 3D CNN achieves an RMSE of 0.4808, while the CNN–VAN–Transformer yields an RMSE of 0.7128. On 90-frame segments, the CNN–VAN–Transformer provides the best accuracy with an RMSE of 0.6946, compared with 0.8894 for the 3D CNN; V1-3DCNN degrades to 1.5828, indicating that longer context does not guarantee improvement and that window–architecture interaction is substantial. Video-level MAE varies by content and target range: for example, the best MAE occurs at 120 frames for HEM022 (0.350) and HEM188 (0.150), at 60 frames for HEM086 (0.161), and at 30 frames for HEM028 (0.512), highlighting the absence of a universal optimum window length across subjects/videos.

Conclusion—Edge measurements on a Raspberry Pi 5 using an ACS711 current sensor and microcontroller-triggered logging show a monotonic energy increase with window size: mean power rises from 2.481 W (30-frame) to 3.951 W (120-frame), and mean energy from 1.029 J to 5.047 J. The maximum measured energy is 7.14 J (HEM022, 120-frame), while the minimum is 0.48 J (HEM028, 30-frame); peak power reaches 6.80 W in the 120-frame setting. These findings demonstrate that sliding-window length is not merely a tuning knob but a primary design decision that directly balances accuracy, output stability, and energy budget in practical, deployable SpHb estimation systems.

Key Words—Contactless hemoglobin estimation, Facial video, Sliding window, Edge computing, Energy consumption.

This article published under the Creative Commons Attribution-NonCommercial-ShareAlike 4.0 (CC BY-NC-SA) licence.

I. INTRODUCTION

The spectrophotometric peripheral hemoglobin (SpHb) is a clinically meaningful surrogate for hemoglobin status and is widely used for anemia screening, monitoring suspected blood loss, supporting transfusion decisions, and evaluating treatment response. In routine practice, hemoglobin assessment is typically obtained via invasive, laboratory-based measurements. However, invasive sampling imposes constraints related to consumables, infrastructure, time, cost, measurement frequency, and feasibility under field conditions. These limitations become more pronounced in disaster and crisis scenarios, where rapid point-of-need decision-making is required; consequently, contactless and low-cost approaches become even more important.

Motivated by these needs, the literature has increasingly emphasized non-invasive, smartphone- and image-based approaches for hemoglobin/anemia estimation. HemaApp is an early example that targets hemoglobin estimation using optical measurement principles at the fingertip with a smartphone camera and illumination sources [1]. Approaches that estimate hemoglobin and screen for anemia from nail-bed photographs further support the concept of rapid self-screening using user-acquired images [2]. In the emergency department context, studies predicting anemia from facial images using deep learning have demonstrated the potential for fast screening within clinical decision workflows [3]. More recently, in an effort to bring hemoglobin estimation closer to real-world clinical practice, models that also consider the clinical data/workflow context and are designed for integration into decision processes have been reported [4]. Nevertheless, a substantial portion of prior work is either classification-oriented or performs inference from a single image. In contrast, continuous-valued SpHb regression and the ability to produce stable outputs over time offer higher utility for monitoring and decision support.

In this respect, facial video provides an additional advantage over still images. Beyond spatial cues, video captures subtle temporal variations in color, thereby enabling richer spatiotemporal representations. Indeed, in tasks such as remote photoplethysmography, lightweight 3D CNN approaches capable of real-time operation have been shown to improve robustness to motion and noise by explicitly modeling the temporal dimension [5]. Focusing directly on contactless hemoglobin estimation, Bal et al. established a strong foundation by demonstrating SpHb regression from facial videos using 3D CNN-based models [6]. However, translating this line of research into deployable products and systems requires more than reporting predictive performance metrics. Practical deployment must also account for system-level constraints—notably real-time feasibility on edge devices, energy consumption, and latency.

The literature on real-time video analytics discusses these requirements extensively within the framework of edge computing. To reduce bandwidth demand and lower latency through near-source processing, resource-efficient infrastructures and strategies such as frame reduction and early filtering have been proposed for edge video surveillance [7]. Other studies aim to meet latency and energy targets by partitioning deep-network inference between the device and an edge server, yielding notable benefits in resource-constrained IoT settings [8]. Optimizing the accuracy–computation–bandwidth trade-off by jointly considering the encoding, optimization, and inference pipeline in mobile edge networks under harsh conditions is another active research axis in edge video analytics [9]. In addition, energy awareness can be strengthened in real-time video tasks through pruning/quantization and hardware-specific optimizations [10] , and low-power accelerators on single-board computers (e.g., Raspberry Pi, Jetson Nano) have been reported to improve energy efficiency [11]. Edge-based architectures have also been shown to satisfy real-time requirements in facial analytics scenarios [12]. Overall, existing edge video analytics research primarily focuses on infrastructure design, task partitioning, and optimization under bandwidth/energy constraints for tasks such as object detection, tracking, and surveillance [7]–[9]. By contrast, the contactless hemoglobin/SpHb estimation literature, while demonstrating model accuracy, has generally not reported on-device latency–energy–sustained inference behavior in a systematic manner, nor has it examined in depth how key parameters in video-based regression—such as sliding-window length—affect both output stability and computational cost [6]. This gap becomes particularly critical in disaster/crisis contexts, where cloud connectivity may be intermittent, energy resources may be limited, and rapid triage demands may increase. In such environments, the ability to run the decision-support algorithm on-device and to assess output stability reliably are central determinants of real-world value.

In this paper, we address contactless SpHb estimation from facial videos from an edge-computing perspective. Our goal is to quantify the feasibility of real-time inference on an edge device and to provide a holistic analysis of how the sliding-window length influences prediction accuracy, prediction dynamics, and stability over the course of a video, latency, and energy consumption. The main contributions are as follows:

- We deploy the contactless SpHb estimation pipeline on an edge device and measure latency and energy, enabling a quantitative discussion of field deployability.
- We train and compare models using window lengths spanning 30–120 frames and demonstrate the accuracy–cost trade-off.
- We present a comparative analysis of time-resolved predictions produced by different window configurations over entire video sequences, moving beyond a single RMSE value to include output stability and deviation patterns.
- Considering cloud dependence and communication/energy constraints, we highlight the importance of an edge approach for contactless SpHb estimation in austere settings.

II. MATERIAL and METHOD

This study comprises two main components. First, we develop a video-regression pipeline that estimates contactless total hemoglobin, denoted as SpHb, from facial video. Second, we implement a measurement-driven edge system that quantifies latency and power consumption while running real-time inference on an edge device. Figure 1 illustrates how raw facial videos and their hemoglobin labels are transformed into a standardized training-ready data package. Figure 2 depicts the power measurement setup used during on-device inference on a single-board computer (Raspberry Pi 5), including the current sensor and microcontroller. In this setup, the current sensor connected in-line with the single-board computer's power supply continuously streams current measurements to the microcontroller. When the algorithm starts executing on the single-board computer, a trigger pin is set high to initiate recording on the microcontroller, and the measurement stream

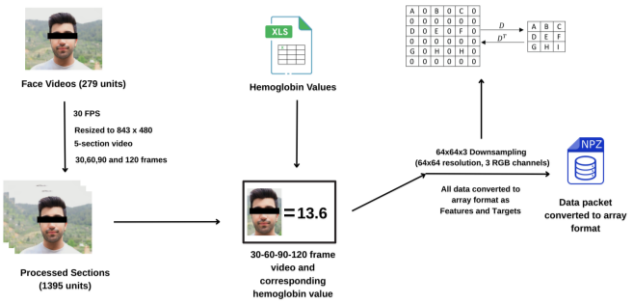


Fig. 1. Pre-processing pipeline for SpHb estimation.

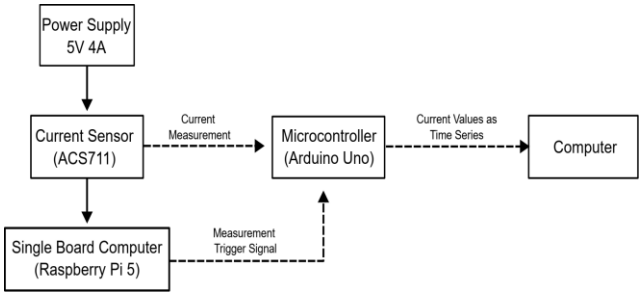


Fig. 2. Single-board computer and current measurement setup.

is transmitted to a host computer for storage. When the program terminates, the trigger pin is pulled low, thereby ending the measurement session. The measurement system records at a sampling rate of 50 Hz. In the context of edge computing, executing video analytics close to the data source is preferred because it reduces network dependence and enables more sustainable solutions under latency and energy constraints [7]–[9]. For contactless hemoglobin estimation, video-based regression can provide a stronger representation than still-image inference because it incorporates temporal information [6].

The dataset consists of 279 facial videos and a single paired target value for SpHb (g/dL) per recording. The distribution of SpHb levels in the dataset is presented in Fig. 3. Each video was processed at 30 FPS, and the SpHb value for a given recording was propagated as the label for all segments derived from that video.

The data preparation pipeline, as summarized in Fig. 1, operates as follows. First, each video is decomposed into frames at 30 FPS. Next, to balance computational load and standardize input size, frames are resized to 843×480. In this study, the train/validation/test split is performed subject-independently: images from the same individual are never distributed across different subsets, and all videos from each participant are assigned to a single subset. This strategy eliminates the risk of data leakage that could arise if samples derived from the same person appeared in multiple splits, thereby ensuring that the reported performance more reliably reflects real-world generalizability.

To increase the number of training samples and enhance within-video temporal diversity, each video is divided into five segments, yielding a total of 1,395 processed segments. Fixed-length video clips are then generated from each segment using a sliding-window approach. We define four window-length configurations—30, 60, 90, and 120 frames—and treat window length as an explicit design parameter. To reduce inference cost on the edge device, each clip is downsampled to 64×64×3 (RGB) and converted into the model's input tensor. Finally, each sample is packaged as an input video tensor paired with its corresponding target value.

The SpHb estimation task is formulated in this study as single-output, continuous-valued regression, rather than classification. Accordingly, the final layer of all models is designed as a regression head consisting of a single neuron with linear activation. We compare three model families. The first family comprises 3D convolutional neural networks (3D CNNs), which can jointly process spatial and temporal dimensions. 3D CNNs are a well-established baseline for spatiotemporal feature learning in video analytics [13] , and their direct applicability to contactless SpHb regression has also been demonstrated [6]. The second family consists of hybrid architectures that strengthen a CNN-based backbone with attention mechanisms and subsequently employ a Transformer to model global context. Owing to its capacity to capture long-range dependencies, the Transformer architecture has become a standard component in many visual tasks [14]. In this work, the CNN–VAN–Transformer approach is built on the Visual Attention Network (VAN) concept, which aims

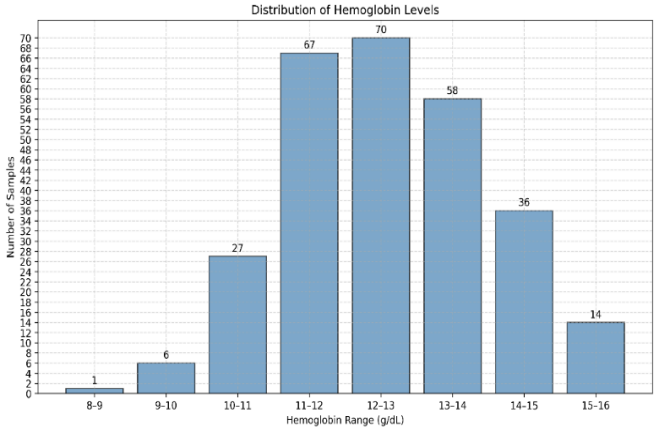


Fig. 3. SpHb distribution in the dataset.

to achieve efficient representation learning through visual attention mechanisms [15]. The third family is the V1-3DCNN variant, incorporated to assess how variations in model complexity influence both accuracy and edge-related costs.

These architectures are also conceptually aligned with modern Transformer-based directions for video representation learning. For instance, the Video Swin Transformer provides a strong reference framework for video tasks by balancing inductive locality bias with computational cost [16]. Since real-time inference on mobile/edge platforms is critical for making hemoglobin estimation practical in field and crisis conditions, the motivation for real-time hemoglobin estimation on smartphones or portable devices is consistent with the design objectives of this study [4].

Training is conducted using mini-batch optimization, and early stopping based on validation performance is applied to mitigate overfitting. Batch size is selected based on the memory requirements imposed by the window length and architectural complexity, enabling a practical comparison across different temporal-window configurations. The data split strictly separates training, validation, and test sets; model selection is performed on the validation set, and final reporting is conducted only on the test set. This protocol is a standard requirement for reliable generalization assessment in video-based prediction studies [6].

For algorithmic evaluation, RMSE and MAE are used as regression metrics. RMSE penalizes larger deviations more strongly due to the squared error term, whereas MAE provides a more directly interpretable measure of average absolute deviation. However, for real-time video-based systems, single scalar metrics are insufficient because the model produces sequential predictions under a sliding-window regime, and the temporal stability of these predictions is critical for practical reliability. Therefore, we perform an output-stability analysis: for selected videos, we compare how predictions evolve under different window lengths. This enables a unified assessment of how increasing window length can smooth and stabilize predictions while simultaneously increasing latency and energy cost.

Edge-performance evaluation reports two groups of measures are inference latency and power/energy consumption. Inference latency is the time (ms) required for the model to produce a prediction per window/clip on the edge device and is a direct indicator of real-time feasibility. Power/energy measurement follows the hardware topology in Fig. 2. The instantaneous current $I(t)$ is measured through an ACS711 current sensor connected in series with the Raspberry Pi 5's 5 V supply line, and an Arduino UNO samples the sensor output and transmits it to the Raspberry Pi via serial communication. On the Raspberry Pi side, measurements are logged with timestamps, and power is

calculated by $P(t) = V \cdot I(t)$, with $V = 5\text{V}$ assumed constant. Total energy is then obtained over discrete time steps using (1):

$$E = \sum_k P_k \cdot \Delta t_k. \quad (1)$$

For power consumption, summary statistics such as mean, maximum, minimum, and standard deviation are reported, while energy consumption is noted as the accumulated value in Joules. Prior works on Raspberry Pi–based workloads commonly use standardized power-efficiency metrics and reports how accelerators and optimizations affect energy consumption. Following the same methodology, we jointly evaluate accuracy (RMSE/MAE), output stability (time-series behavior), latency, and energy consumption. This allows us to analyze not only the best predictive accuracy, but also the accuracy–stability–energy trade-off required for real-world deployability.

III. RESULTS and DISCUSSION

In this section, the experimental results of the proposed facial video–based SpHb estimation approach—developed under the Materials and Methods framework—are jointly evaluated in terms of both algorithmic accuracy (RMSE, MAE) and edge deployability (power/energy consumption). The findings are reported through learning curves that characterize training dynamics, comparative performance analyses across different window lengths, time-series prediction behavior on representative videos, and power/energy measurements collected on a single-board computer. The objective is not merely to minimize error, but to identify a field-deployable operating point by balancing window length (sliding window), accuracy, output stability, and energy budget.

A. Training dynamics and model convergence

Figure 4(a), which summarizes the training process of the 3D CNN architecture, shows that both the training loss and training RMSE decrease steadily as epochs progress, whereas the validation loss and validation RMSE exhibit intermittent spikes. This pattern indicates that the model increasingly fits the training data, while the error on validation samples fluctuates with higher variance. In video-based regression problems, such behavior can be attributed to factors such as illumination and pose variations, fluctuations in facial ROI quality, and potential label noise arising from assigning a single SpHb label to multiple temporal windows extracted from the same video. Accordingly, the evidence in Fig. 4(a) suggests that generalization performance is shaped not only by architectural choices but also by the data-splitting strategy—particularly subject-level separation—and by regularization mechanisms such as early stopping, dropout, and weight decay. Figures 4(b) and 4(c) similarly show a decreasing trend in training error, but their validation curves exhibit more pronounced oscillations. Notably, the validation RMSE does not stabilize even after incorporating attention or RNN blocks, suggesting that temporal modeling alone is insufficient; instead, dataset heterogeneity and the windowing strategy strongly influence inference performance. In addition, the training/validation MAE curve in Fig. 5 shows that MAE drops rapidly during the initial epochs, then plateaus. This indicates diminishing marginal gains as training proceeds, supporting early stopping as an appropriate strategy to improve both generalization and training efficiency.

B. Architectural performance comparison across window lengths

In this section, we investigate how the sliding-window length (30, 60, 90, and 120 frames) influences both architectural choice and overall performance in facial video–based SpHb estimation. Throughout the experiments, the data preparation pipeline remains constant; only the segment length generated by the sliding window is varied. For each window length, multiple deep learning architectures are compared under a standard evaluation

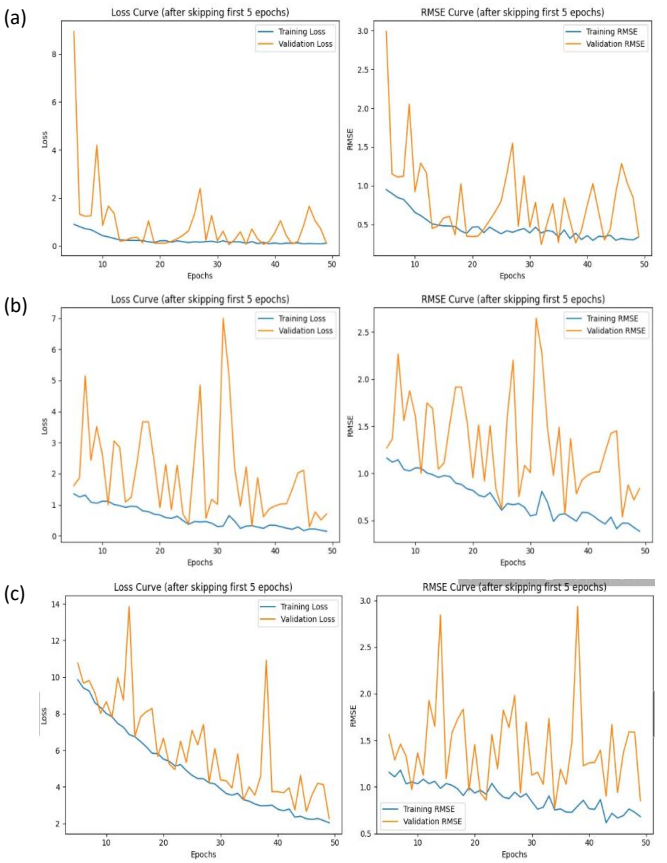


Fig. 4. Loss and RMSE Curves: (a) 3D CNN Model, (b) CNN–VAN–Transformer Model, (c) V1-3DCNN Model.

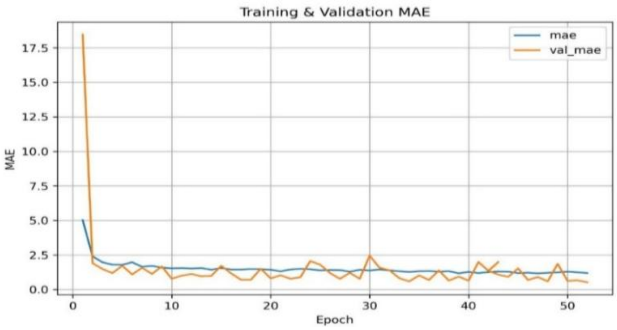


Fig. 5. Training and validation MAE Curves for the 3D CNN model.

protocol, and performance is reported as RMSE. This analysis not only examines whether a longer temporal context improves accuracy but also reveals the extent to which different architectures can effectively exploit it.

The results for 30-, 60-, 90-, and 120-frame settings are reported in Table 1. A short window is a natural candidate in edge scenarios due to its potential for lower latency; however, the limited temporal context may cause physiological cues to be overwhelmed by noise. As shown in Table 1, under 30-frame setting, the 3D CNN approach achieves the lowest error with RMSE ≈ 0.6701, while the CNN–VAN–Transformer performs at a comparable level with RMSE ≈ 0.6886. In contrast, the substantially weaker performance of V1-3DCNN under the same condition (RMSE ≈ 1.190) suggests that specific 3D variants may fail to reach sufficient generalization capacity when the temporal window is short.

Compared to the shorter window, the 60-frame sliding-window setting provides a richer temporal context, allowing physiological signal components to accumulate while partially averaging out noise. The effect of this increased context is evident in Table 1: the 3D CNN model achieves the lowest error with RMSE ≈ 0.4808,

Table 1. Performance Comparison of Different Deep Learning Architectures on 30-, 60-, 90-, and 120-Frame Video Segments.

Model	Frames	Epochs	Batch Size	Mean	Std. Dev.	RMSE
3D CNN	30	46 (ES)	16	13.85	0.123	0.670
	60	72 (ES)	8	10.644	0.069	0.481
	90	90	23 (ES)	8	13.438	0.111
	120	100	8	13.365	0.205	1.32
CNN VAN Transform	30	24 (ES)	8	14.004	0.143	0.689
	60	26 (ES)	8	10.899	0.101	0.713
	90	39 (ES)	8	13.231	0.073	0.695
	120	100	8	13.769	0.156	0.99
V1-3DCNN	30	100	8	14.642	0.270	1.190
	60	100	8	11.847	0.194	1.674
	90	100	8	13.947	0.145	1.583
	120	100	8	13.969	0.123	0.64

whereas the CNN–VAN–Transformer architecture exhibits a more limited performance with RMSE ≈ 0.7128. These findings indicate that a 60-frame window creates a more favorable temporal context than the short-window regime; however, architectural choice remains a decisive factor. Accordingly, while 60 frames is a plausible candidate for a "reasonable midpoint" between accuracy and latency, the selected architecture still directly determines the overall system performance.

In the 90-frame case, the CNN–VAN–Transformer architecture achieves the lowest error, with RMSE ≈ 0.6946, whereas the 3D CNN approach exhibits a higher error, with RMSE ≈ 0.8894. The increase of V1-3DCNN to RMSE ≈ 1.5828 further indicates that simply enlarging the temporal window does not guarantee improvement; rather, the architecture–window interaction is substantial. While a 90-frame context can increase the amount of accumulated temporal information and may reduce errors for specific architectures, it also increases computational load on the edge device and, consequently, is expected to raise power/energy consumption.

In the case of 120-frame sliding-window configuration, contrary to expectation, further increasing the window length does not improve performance across all architectures; instead, a notable degradation is observed for some models. For example, the increase to RMSE ≈ 1.32 for the 3D CNN clearly indicates that a longer temporal context is not always beneficial. This outcome can be interpreted from two complementary perspectives. First, with a longer window, variations in illumination, pose, or motion may accumulate over time, introducing unnecessary variance into the input and thereby impairing regression performance. Second, longer windows increase computational cost and can reduce operational efficiency under edge-device constraints; hence, the practical value of selecting a "long window" may diminish in real deployments. In the case of the 120-frame sliding-window configuration, the comparatively better result obtained by V1-3DCNN (RMSE ≈ 0.64) suggests that the architecture–window interaction is not linear and that some architectures may benefit more from extended temporal context. Nevertheless, the overall pattern indicates that moving to 120 frames does not yield a consistent improvement in accuracy.

When four frame configurations given in Table 1 are considered jointly, it becomes clear that window length is not merely a hyperparameter for SpHb estimation, but a critical variable that directly shapes the overall system design. Although a short window offers the advantage of low latency, it can also make accuracy more fragile. In this dataset, window lengths in the 60–90 frame range yield noticeable improvements in accuracy. In contrast, moving to 120 frames increases cost while failing to deliver a guaranteed gain in accuracy.

C. Time-series prediction behavior and output stability on representative videos

To understand field-facing behavior, it is essential to present predictions as time series over representative videos under 30/60/90/120-frame windows. In this section, we analyze prediction traces on selected examples. These videos are not chosen at random; instead, they are selected based on the SpHb distribution in the dataset. Specifically, samples are drawn from the low range (8–11 g/dL), the dense region (11–14 g/dL), and the high range (14–16 g/dL) as indicated by the distribution in Fig. 3. This selection allows us to observe not only performance in the most frequent range, but also how the models behave at the distribution tails where fewer samples are available—namely, how window length anchors the prediction band, how stable the output remains over time, and how predictions respond to changes. This perspective is particularly important for edge scenarios involving continuous reporting, where output stability is a key requirement [11], [17].

Fig. 6(a) shows the time-series predictions produced for the HEM022 video under different sliding-window settings. The ground-truth label for HEM022 is 9.9 g/dL, which falls within the low SpHb range (8–11 g/dL). The plot indicates that the 30-frame model remains in the 13–14 g/dL band throughout the video, substantially overestimating the actual value. This suggests that a short window may not only increase variability but can also introduce a strong positive offset in the output level. Under 60- and 120-frame windows, predictions approach approximately 10.5–11.0 g/dL, whereas the 90-frame model appears to settle in a higher band (around 12.6 g/dL). A key observation here is that, across window lengths, not only the magnitude of error but also the direction of bias changes; thus, in the low SpHb region, window selection can directly affect calibration requirements in practical deployment.

For HEM028, the ground-truth SpHb is 13.5 g/dL, which falls within the dense 11–14 g/dL range. The behavior of different window settings in this video is illustrated in Fig. 6(b). Here, the 30-frame model tends to remain mostly above 14 g/dL, indicating that the short window can produce an upward offset even in the mid-range. The 60- and 90-frame models, in contrast, track lower bands (approximately 12.6–13.0 g/dL) and thus underestimate the actual value. The 120-frame model produces a band closer to the 13.5 g/dL reference line but exhibits a more fluctuating profile over time. This example shows that there is no universally optimal window even in the dense region; window length can influence proximity and stability in different, sometimes competing, ways.

The example in Fig. 6(c) (HEM044) represents the high SpHb region (14–16 g/dL), with a ground-truth value of 15.7 g/dL, located in the right tail of the distribution. The plot indicates that, across all window settings, predictions remain clearly below the actual value. In particular, the 60-frame model stays relatively stable around 12.5–13.0 g/dL, while the 90-frame model remains stable around 13.4–13.6 g/dL. The 120-frame model rises initially and then shifts downward to approximately 12.5 g/dL, indicating a band change over time. This behavior suggests a potential generalization issue in the high SpHb regime, consistent with a tendency toward regression to the mean. In other words,

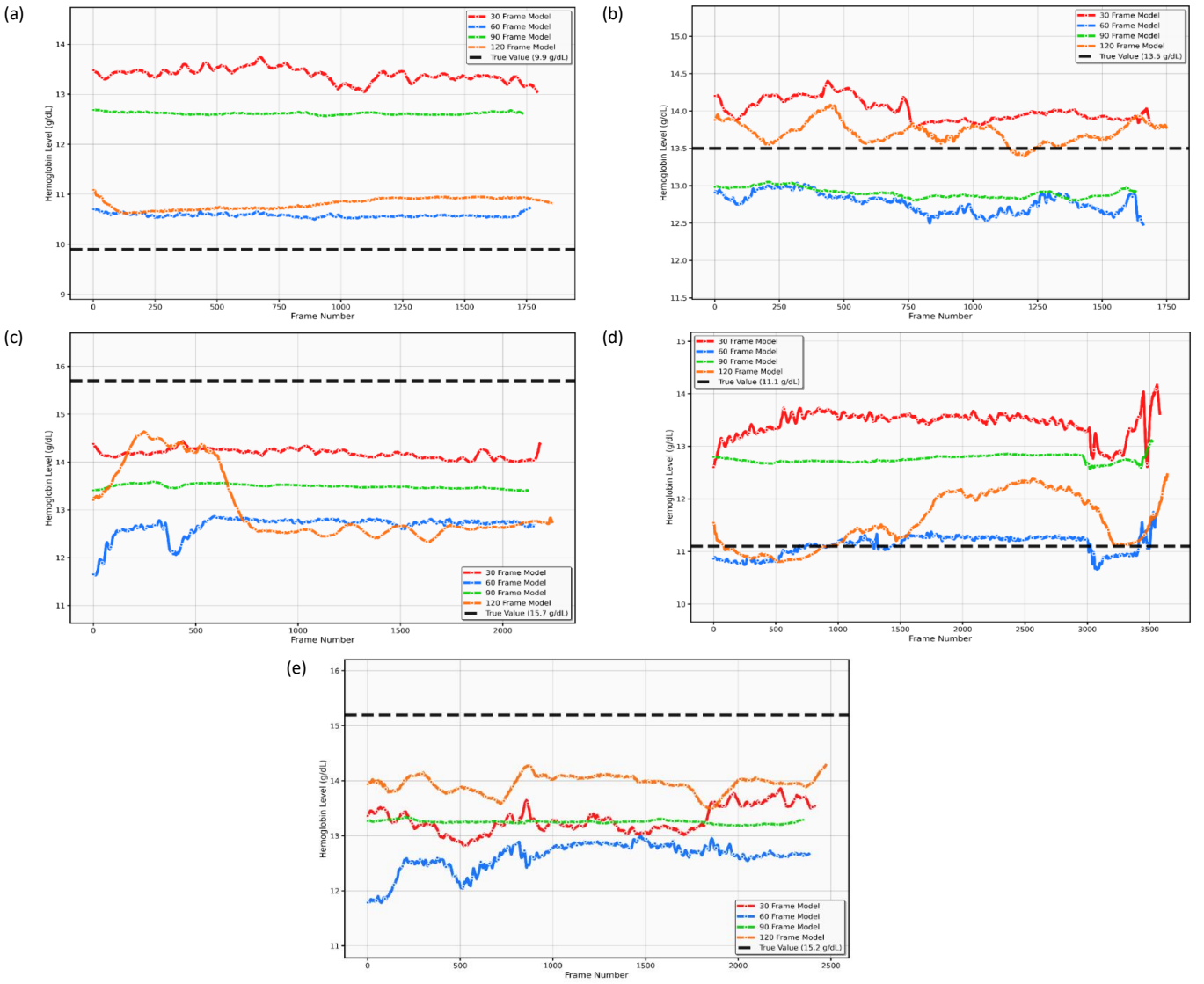


Fig. 6. SpHb Prediction Comparison for five sample videos: (a) HEM022 video, (b) HEM028 video, (c) HEM044 video, (d) HEM086 video, (e) HEM188 video.

the relatively small number of samples in the right tail may lead the model to produce a systematic negative bias in this clinically significant range. This finding motivates explicitly reporting high-range performance and, if necessary, adopting training strategies that better balance this range.

The HEM086 video has a ground-truth SpHb of 11.1 g/dL, positioned near the lower boundary of the dense region, and its longer duration strengthens the discussion of temporal stability. Fig. 6(d) presents the time-series behavior for different window settings on HEM086. The 60-frame model remains within a relatively narrow band around the 11.1 g/dL reference line for an extended period. In contrast, the 30-frame model stays in a higher band above 13 g/dL and produces noticeable jumps toward the end. The 120-frame model exhibits a gradual upward trend and changes bands in certain intervals, suggesting that long windows may accumulate "trend" and respond more slowly to content changes. This example supports the following implication: if continuous reporting is required in field deployment, window selection should be evaluated not only by error metrics but also for robustness against drift and abrupt deviations.

Finally, Fig. 6(e) shows sliding-window predictions for HEM188, a second high-range example with a ground-truth SpHb of 15.2 g/dL. The 60-frame model stays in the 12.6–12.9 g/dL band, the 90-frame model remains highly stable around 13.2 g/dL, and the 30-frame model fluctuates more within approximately 13.3–13.7 g/dL. The 120-frame model increases over time and occasionally approaches the 14 g/dL band, suggesting that a long window can push predictions upward in this case; however, the predictions remain clearly below the reference line. Overall, HEM188 highlights that in the high SpHb region, window choice often shifts from improving accuracy to determining which band the prediction stabilizes in.

Table 2 summarizes the MAE values obtained for five representative videos under sliding-window lengths of 30/60/90/120 frames. The table clearly shows that no single window length consistently yields the lowest error across all videos. For instance, for HEM022, the 120-frame window achieves the lowest MAE, whereas for HEM086, the 60-frame window produces a markedly lower MAE; for HEM028, the short window (30 frames) can be relatively more advantageous. This finding indicates that window selection is not governed by a universal optimum but is instead sensitive to video content and the target-value range. In other words, even a purely tabular comparison suggests that fixing a single window length may introduce unnecessary error increases for some cases, while an appropriate window choice can yield tangible accuracy gains.

Table 2. Mean absolute error summary for five sample videos.

Video	30-frame	60-frame	90-frame	120-frame
HEM022	3.495	0.679	2.700	0.350
HEM028	0.512	0.726	0.600	0.850
HEM044	1.521	3.028	2.300	0.400
HEM086	2.336	0.161	1.700	2.200
HEM188	1.912	2.585	2.100	0.150

When the quantitative error values in Table 2 are examined together with the band behavior observed in the time-series plots, it becomes evident that sliding-window length affects model behavior not only through average error, but also through temporal consistency, the direction and magnitude of systematic bias, and the responsiveness to changes. Accordingly, window-length selection should be treated not merely as a hyperparameter setting, but as a design decision that directly determines output quality and reporting reliability for an edge-deployed system. These results also suggest that a more suitable approach may be to dynamically select the window length, guided by signals such as video quality, motion level, or ROI reliability. At minimum, for practical deployment, it is more rational to consider 60–90 frames as an initial operating range rather than relying on a single fixed window.

D. Power and energy measurements on the edge device

In this section, we quantitatively characterize the operational cost of the SpHb estimation system in an edge environment. We experimentally examine how sliding-window length affects power and energy consumption. Measurements are performed using an in-line current sensor connected in series with the Raspberry Pi 5 power supply and a microcontroller-based, trigger-driven logging infrastructure. This design targets only the time interval during which the algorithm is running and prevents idle consumption from contaminating the measurements. For each sample video, we report average power, peak power, minimum power, power standard deviation, and total energy (Joules).

Power/energy measurements corresponding to window lengths of 30, 60, 90, and 120 frames are presented in Table 3. When these tables are considered together, a consistent trend emerges as window length increases, both the average power and the resulting total energy consumption generally increase. The highest total energy consumption is observed for the HEM022 sample video under the 120-frame condition, measured as 7.14 J. In contrast, the lowest total energy consumption is reported for HEM028 under the 30-frame condition, at 0.48 J. A similar pattern is observed for peak power: the highest value again occurs in the 120-frame scenario for HEM022, reaching 6.80 W, whereas shorter windows yield comparatively lower peak power levels.

The increase in energy cost is supported not only by individual examples but also by window-length–level summary statistics. Table 4, which aggregates the power/energy metrics by window length, presents this trend more compactly. The mean power rises from approximately 2.48 W for the 30-frame condition to 2.75 W (60 frames), 3.45 W (90 frames), and 3.95 W (120 frames). The same trend is observed for total energy: Table 4 reports an average energy of ≈1.03 J for 30 frames, ≈2.96 J for 60 frames, ≈4.18 J for 90 frames, and ≈5.05 J for 120 frames. These results indicate that longer windows increase workload, which directly translates into higher energy consumption on the edge device. Overall, sliding-window length is a system-design parameter that jointly determines accuracy and energy budget. Short-to-medium windows offer a practical balance for battery-constrained field scenarios, whereas longer windows may be justified when the energy budget permits and accuracy is the priority.

A. Comparison with related work

In literature, studies examining deep learning on edge platforms are organized mainly around three axes: FPS/latency, model optimization, and system architecture. For example, work analyzing real-time DNN inference on single-board computers has highlighted latency and efficiency bottlenecks in CPU-based execution and discussed core benchmarking criteria for edge settings [17]. More recent studies have systematized power measurement; for instance, energy profiling on a Raspberry Pi 4B has been reported using power meters with per-second current/voltage logging, and comparisons between accelerators (e.g., NCS2) and CPU execution have strengthened the perspective of power efficiency, rather than focusing solely on speed [11]. On the video analytics side, edge–cloud partitioning has been studied jointly with bandwidth and latency constraints, emphasizing the tension between the energy cost of running the full model on-device and the communication cost of offloading [7],[8]. In contrast, the present study targets the specific problem of contactless SpHb estimation and contributes along three tightly integrated dimensions. First, for spatiotemporal video-based regression, we systematically sweep the sliding-window length. Second, we report predictive performance (RMSE/MAE) together with video-level time-series behavior, enabling assessment beyond aggregate error values. Third, under the same experimental conditions, we directly measure and report the edge device's power and energy profile. This unified reporting is distinctive in that it brings together machine-learning performance and power/energy characteristics—dimensions that are often treated separately in prior work—within a single experimental workflow [11], [17]. In this broader context, fog- and edge-oriented computing architectures have been associated with improved operational efficiency under resource-constrained conditions, particularly with respect to processing locality, system responsiveness, and energy-aware design [18], [19]. Recent studies on edge-AI-enabled IoT systems

Table 3. Power and energy measurements for SpHb estimation on 30-, 60-, 90-, and 120-frame video windows.

Video Name	Frame	Video Length (sec)	SpHb	Mean Pow.	Max Pow.	Std. Dev.	Min Pow.	Energy (J)
HEM022	30	66	9.9	2.13	4.15	1.14	0.85	1.60
	60	66	9.9	2.81	4.15	0.85	1.40	4.23
	90	66	9.9	3.74	5.70	1.27	1.95	5.93
	120	66	9.9	4.41	6.80	1.56	2.35	7.14
HEM028	30	62	13.5	1.27	1.40	0.19	1.00	0.48
	60	62	13.5	1.85	3.60	1.02	0.85	2.06
	90	62	13.5	3.18	5.13	1.17	1.82	4.33
	120	62	13.5	4.13	6.21	1.28	2.51	5.93
HEM044	30	79	15.7	2.92	3.60	0.53	2.30	1.04
	60	79	15.7	3.04	3.60	0.49	2.30	3.20
	90	79	15.7	4.12	5.46	0.95	2.45	5.28
	120	79	15.7	4.89	6.78	1.28	2.56	6.76
HEM086	30	125	11.1	3.10	3.40	0.42	2.50	1.37
	60	125	11.1	3.28	4.15	0.46	2.85	3.38
	90	125	11.1	3.49	5.77	1.08	2.21	3.84
	120	125	11.1	3.64	6.91	1.51	1.75	4.17
HEM188	30	56	15.2	3.17	3.60	0.32	2.85	1.18
	60	56	15.2	3.09	4.15	0.81	2.10	3.30
	90	56	15.2	3.51	4.88	0.97	2.34	3.85
	120	56	15.2	3.80	5.40	1.09	2.52	4.24

Table 4. Power and energy metrics across different sliding-window lengths.

Window Length	Mean Pow (W)	Max Pow (W)	Std. Dev	Min Pow. (W)	Energy (J)
30-Frame	2.481	2.839	0.359	2.019	1.029
60-Frame	2.751	3.932	0.724	1.969	2.955
90-Frame	3.453	5.362	1.083	2.121	4.180
120-Frame	3.951	6.372	1.337	2.229	5.047

have likewise continued to emphasize latency, computational load, and energy consumption as core determinants of deployability in real-world settings [20]. From this perspective, the present study contributes to literature by examining these constraints within the specific context of contactless SpHb estimation and by directly relating predictive behavior to measured edge-device energy cost. Table 5 compares the reporting scope of this study against selected edge studies for which a direct power–performance comparison is feasible.

IV. CONCLUSION

This study demonstrates that sliding-window length is not merely a hyperparameter but a primary system-level design decision for contactless SpHb estimation on edge devices. Our unified experimental framework—covering RMSE/MAE, time-series output stability, and directly measured power/energy on a Raspberry Pi 5—reveals three key findings. First, mid-range windows (60–90 frames) generally provide the best trade-off between accuracy and temporal consistency. Second, increasing the window to 120 frames raises energy cost without guaranteeing accuracy gains, as accumulated within-video variations can degrade model performance. Third, energy measurements confirm a monotonic increase from 1.03 J (30-frame) to 5.05 J (120-frame) per inference cycle, directly constraining battery-powered operating time. These results collectively establish that window-length selection must balance accuracy, stability, and energy budget for field-deployable systems. Energy measurements clearly show that both mean power and total energy consumption increase systematically with window length. Consequently, window-length selection is not merely an algorithmic setting; it is a system-level parameter that directly affects battery-powered operating time, field sustainability, and the device's thermal and operational limits. A central takeaway of this work is that sliding-window length is more than a training-time hyperparameter—it is a design decision that governs the trade-off among accuracy, output stability, and energy budget.

Several research directions can be identified to build upon the findings of this study. First, the systematic biases observed at the distribution tails (low and high SpHb ranges) may be reduced through sample rebalancing and calibration strategies that give more weight to underrepresented hemoglobin levels. Second, the single-label-per-video assumption in this work can be addressed by collecting datasets with continuous or repeated hemoglobin measurements during the recording session, which would allow a more accurate evaluation of temporal prediction behavior. Third, adaptive window selection methods that consider video content characteristics—such as ROI quality, motion level, and illumination conditions—could improve the robustness of predictions across different recording environments. Fourth, energy efficiency on edge devices can be further improved by applying model compression techniques such as quantization and pruning, as well as by integrating hardware accelerators. Finally, external validation on larger datasets that include more diverse populations in terms of age, skin tone, and clinical conditions would provide stronger evidence for the real-world applicability of the proposed approach.

AUTHOR STATEMENT

Plagiarism Check—The article was scanned with iThenticate and found to be compliant with the journal's plagiarism policy.
Conflict of Interest—This article does not have any conflict of interest with any person or organization.
Ethics Committee Approval— This study was performed in line with the principles of the Declaration of Helsinki. Approval was granted by the Ethics Committee of University Manisa Celal Bayar (2024/2515).
Use of Artificial Intelligence Tools—In this study, no artificial intelligence tools were used in article writing. All content reflects the original contribution of the author.
Funding—This article was supported under the Scientific and Technological Research Council of Türkiye, TÜBİTAK-2209A project numbered 1919B012417134 and TÜBİTAK-1001 under Project No. 123E689.
Data Availability—Data is available upon request but restricted due to ethical or privacy reasons.

Table 5. Comparison with related works.

Ref.	Task	Hardware	Power/Energy Measurement	Reported Metrics	Overlap / Difference Relative to This Study
[17]	DNN inference benchmarking	Raspberry Pi–a single-board computer	Primarily latency- and inference-focused; energy is discussed mainly in a benchmarking context.	Latency, throughput	Provides a foundational edge benchmarking framework; however, it does not address hemoglobin estimation and does not analyze sliding-window length
[11]	Image recognition and tracking	Raspberry Pi 4B and NCS2	External power meter with per-second logging; includes buffering overheads.	FPS, power profile, accelerator comparison	Similar measurement methodology; this study additionally evaluates sliding-window configurations and reports SpHb time-series prediction behavior
[8]	Distributed inference and video surveillance	IoT device and edge server	Targets energy efficiency via model splitting/offloading strategies	Latency, workload reduction, energy savings	Shares the energy–latency trade-off perspective; does not consider hemoglobin regression or window-length effects.
[7]	Edge infrastructure for video analytics	Edge layer and end devices	Efficiency- and bandwidth-oriented optimization	Bandwidth, latency, resource utilization	Strong system-level optimization focus; this study instead provides deeper single-device profiling of inference and directly measured power and energy
[12]	Facial expression recognition	Raspberry Pi	Motivated by edge deployment and latency constraints	Accuracy, real-time performance	Uses facial video similarly; the target task is not hemoglobin, and energy measurement is not reported systematically.
[10]	Object detection optimizations	Edge- and IoT-oriented platforms	Emphasizes resource constraints via pruning and quantization	Accuracy–resource trade-offs	Shares an optimization-oriented viewpoint; this study provides concrete benchmarking via window-length analysis and direct power measurement.
This study	Healthcare-oriented video regression (SpHb)	Raspberry Pi 5, ACS711, and Arduino (triggered logging)	Time-stamped current time series with triggered acquisition to isolate active inference intervals	RMSE, MAE, and energy (J)	Sliding-window length (30/60/90/120 frames) is the primary control variable governing the accuracy–stability–energy trade-off.

REFERENCES

[1] E. J. Wang, W. Li, D. Hawkins, T. Gernsheimer, C. Norby-Slycord, S. N. Patel, "HemaApp: noninvasive blood screening of hemoglobin using smartphone cameras", *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp)*, Heidelberg, Germany, 12-16 September 2016, 593–604. https://doi.org/10.1145/2971648.2971653.

[2] R. G. Mannino, D. R. Myers, E. A. Tyburski, C. Caruso, J. Boudreaux, T. Leong, G. D. Clifford, W. A. Lam, "Smartphone app for non-invasive detection of anemia using only patient-sourced photos", *Nature Communications*, 9, 4924, 2018. https://doi.org/10.1038/s41467-018-07262-2.

[3] A. X. Zhang, J. J. Lou, Z. J. Pan, J. Q. Luo, X. M. Zhang, H. Zhang, J. P. Li, L. L. Wang, X. Cui, B. Ji, L. Chen, "Prediction of anemia using facial images and deep learning technology in the emergency department", *Frontiers in Public Health*, 10, 964385, 2022. https://doi.org/10.3389/fpubh.2022.964385.

[4] Y. W. Chen, X. Y. Hu, Y. Z. T. Zhu, X. Liu, B. Yi, "Real-time non-invasive hemoglobin prediction using deep learning-enabled smartphone imaging", *BMC Medical Informatics and Decision Making*, 24(1), 187, 2024. https://doi.org/10.1186/s12911-024-02585-1.

[5] D. Botina-Monsalve, Y. Benezeth, J. Miteran, "RTrPPG: An ultra light 3DCNN for real-time remote photoplethysmography", *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, New Orleans, LA, USA, 18-24 June 2022, 2145–2153. https://doi.org/10.1109/CVPRW56347.2022.00233.

[6] U. Bal, F. E. Oguz, K. M. Sunnetci, A. Alkan, A. Bal, E. Akkus, H. Erol, A. Ç. Seçkin, "Estimation of Total Hemoglobin (SpHb) from Facial Videos Using 3D Convolutional Neural Network-Based Regression", *Biosensors-Basel*, 15(8), 485, 2025. https://doi.org/10.3390/bios15080485.

[7] P. P. Kumar, A. Pal, K. Kant, "Resource Efficient Edge Computing Infrastructure for Video Surveillance", *IEEE Transactions on Sustainable Computing*, 7(4), 774–785, 2022. https://doi.org/10.1109/TSUSC.2021.3064245.

[8] M. A. Khan, R. Hamila, A. Erbad, M. Gabbouj, "Distributed Inference in Resource-Constrained IoT for Real-Time Video Surveillance", *IEEE Systems Journal*, 17(1), 1512–1523, 2023. https://doi.org/10.1109/JSYST.2022.3198711.

[9] Y. Y. He, P. Yang, T. Qin, J. W. Hou, N. Zhang, "Joint Encoding and Enhancement for Low-Light Video Analytics in Mobile Edge Networks", *IEEE Transactions on Mobile Computing*, 24(4), 3330–3345, 2025. https://doi.org/10.1109/TMC.2024.3514214.

[10] M. E. Isenkul, "Energy-aware deep learning for real-time video analysis through pruning, quantization, and hardware optimization", *Journal of Real-Time Image Processing*, 22(3), 125, 2025. https://doi.org/10.1007/s11554-025-01703-0.

[11] T. Y. Gao, J. Suto, "Acceleration of Image Classification and Object Tracking by the Intel Neural Compute Stick 2 with Power Efficiency Evaluation on Raspberry Pi 4B", *Sensors*, 25(6), 1794, 2025. https://doi.org/10.3390/s25061794.

[12] J. N. Yang, T. T. Qian, F. Zhang, S. U. Khan, "Real-time facial expression recognition based on edge computing", *IEEE Access*, 9, 76178–76190, 2021. https://doi.org/10.1109/ACCESS.2021.3082641.

[13] D. Tran, L. Bourdev, R. Fergus, L. Torresani, M. Paluri, "Learning Spatiotemporal Features with 3D Convolutional Networks", *Proceedings of the IEEE International Conference on Computer Vision*, Santiago, Chile, 11-18 December 2015, 4489–4497. https://doi.org/10.1109/ICCV.2015.510.

[14] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, I. Polosukhin, "Attention Is All You Need", *31st Annual Conference on Neural Information Processing Systems (NIPS)*, Long Beach, CA, USA, 04-09 December 2017.

[15] M.-H. Guo, C. Z. Lu, Z. N. Liu, M. M. Cheng, S. M. Hu, "Visual attention network", *Computational Visual Media*, 9(4), 733–752, 2023. https://doi.org/10.1007/s41095-023-0364-2.

[16] Z. Liu, J. Ning, Y. Cao, Y. X. Wei, Z. Zhang, S. Lin, H. Hu, "Video Swin Transformer", *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, New Orleans, LA, USA, 18-24 June 2022, 3192–3201. https://doi.org/10.1109/CVPR52688.2022.00320.

[17] D. Velasco-Montero, J. Fernández-Berni, R. Carmona-Galán, A. Rodríguez-Vázquez, "Performance analysis of real-time DNN inference on Raspberry Pi", *Proceedings of the Conference on Real-Time Image and Video Processing*, Orlando, FL; USA, 16-17 April 2018. https://doi.org/10.1117/12.2309763.

[18] A. Abbas, A. A. Ibrahim, "Energy Optimization of Fog Computing and IoT Application", *Avrupa Bilim ve Teknoloji Dergisi*, Özel Sayı, 472–475, 2020. https://doi.org/10.31590/ejosat.780969.

[19] T. Ercan, "Importance of Edge Computing in Critical Manufacturing Systems: FPGA Implementation", *Avrupa Bilim ve Teknoloji Dergisi*, 43, 41–47, 2022. https://doi.org/10.31590/ejosat.1201855.

[20] R. K. B. Singh, J. K. Dash, K. H. K. Reddy, "The role of Edge-AI in edge enabled IoT systems: A comprehensive performance analysis", *Peer-to-Peer Networking and Applications*, 19(1), 35, 2025. https://doi.org/10.1007/s12083-025-02182-7.