



Research

# A computer vision sensor for the parallelization of actively regulated capillary slug flow microreactors

Anoj Winston Gladius<sup>1</sup> · Jonas A. Mylenbusch<sup>1</sup> · David William Agar<sup>1</sup>

Received: 15 May 2023 / Accepted: 1 September 2023

Published online: 11 September 2023

© The Author(s) 2023 [OPEN](#)

## Abstract

In this work, a computer vision sensor for the extraction of slug length, slug velocity and phase ratio from capillary liquid–liquid slug flows from video feeds in real-time, including the necessary post-processing algorithms, is developed. The developed sensor is shown to be capable of simultaneously monitoring multiple capillaries and provides reasonable accuracy at less than 3.5% mean relative error. Subsequently, the sensor is used for the control of a parallelized and actively regulated dual-channel slug flow capillary microreactor setup. As a model reaction, the solvent-free epoxidation of methyl oleate with hydrogen peroxide and a phase-transfer catalyst based on tungstophosphoric acid and a quaternary ammonium salt to yield the product 9,10-epoxystearic acid methyl ester is conducted. A space–time yield of  $0.679 \text{ kg L}^{-1} \text{ h}^{-1}$  is achieved.

## Article highlights

- A computer vision sensor is developed to accurately measure slug characteristics in real-time, facilitating efficient monitoring of multiple capillaries.
- The sensor enables effective control of a dual-channel slug flow capillary microreactor setup, improving operational performance.
- The successful model reaction yields a significant amount of 9,10-epoxystearic acid methyl ester, showcasing the system's high productivity.

**Keywords** Slug flow · Flow sensor · Machine learning · Video imaging · Computer vision · Epoxidation · Methyl oleate · Parallelization · Microreactor

## 1 Introduction and motivation

The energy consumption of the process industry plays a significant role in the total global energy demand [1–3]. Chemical processes are core components of most industries and are the largest consumers of energy in the industrial sector. Considering the global threats emerging

from excessive energy consumption, it is imperative to design and operate efficient processes with significantly reduced environmental footprint. Defined by Ramshaw [4] as “Devising exceedingly compact plant which reduces both the ‘main plant item’ and the installations costs.” and Stankiewicz and Moulijn [5] as “Any chemical engineering development that leads to a substantially smaller,

✉ Anoj Winston Gladius, [anoj.gladius@tu-dortmund.de](mailto:anoj.gladius@tu-dortmund.de) | <sup>1</sup>Laboratory for Chemical Reaction Engineering, Department of Biochemical and Chemical Engineering, TU Dortmund University, Emil-Figge-Straße 66, 44227 Dortmund, Germany.



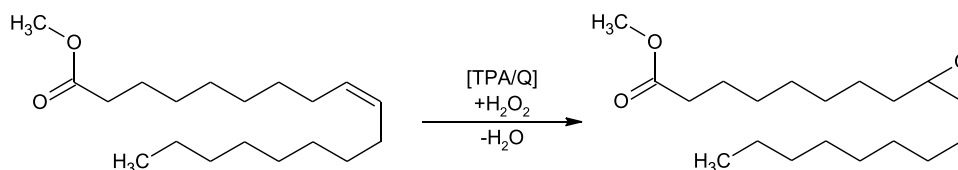
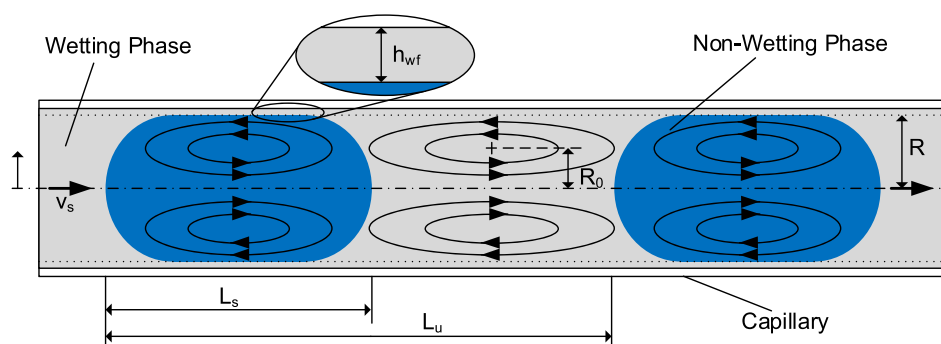
cleaner, and more energy-efficient technology.” Chemical engineers, since the 1970s have slowly began exploring intensified processes to better meet the demands of the ever-increasing global economic engine. This development of chemical processes with emphasis on increased selectivity, space–time yield, and decreased specific energy consumption is commonly referred to as process intensification [6]. One of the four generic principles of Process Intensification according to Gerven and Stankiewicz [7] is the targeted enhancement of mass and heat transfer rates. This is typically achieved in micro-scale equipment, such as microreactors, due to their significant increase in specific interfacial area. Because of this, parameters relevant for efficiency can differ by several orders of magnitude compared to conventional-scale systems [8]. This study encompasses various concepts, notably the advancement of a controlled automated microfluidic reactor system. The system developed in this research exhibits promising prospects as a laboratory tool for conducting high throughput screenings of conditions and for generating valuable products on a smaller–medium scale. These applications exemplify the versatility and significance of slug flow, offering potential contributions to the ongoing process intensification endeavours aimed at enhancing the efficiency and effectiveness of chemical processes.

Capillary slug flow is a multiphase flow pattern characterized by the regular occurrence of capsule-shaped segments, or “slugs”, made up of a disperse phase, separated from each other by liquid segments made up of the continuous phase. The wetting behaviour of each liquid

toward the capillary material determines which phase acts as the continuous phase. The disperse phase is completely enclosed by the continuous phase, which forms a thin liquid film at the tube wall of height  $h_{wff}$ , as presented in Fig. 1. The well-defined flow pattern provides a uniform interfacial area and internal circulation patterns, creating a favourable environment for mass transfer processes between the two phases. This makes the capillary slug flow regime a subject of active research in the intensification of liquid–liquid contactors. It is, for example, well-established that the advantages of the capillary slug flow may be used to intensify mass-transfer-limited liquid–liquid multiphase reactions [9, 10]. Additionally, the use of capillary tubes with an inner diameter of  $\leq 1$  mm provides a high volume-specific surface area, intensifying heat- and mass transfer across the capillary wall [11].

Liquid–liquid capillary slug flows are characterized by three parameters: The slug length  $L_s$ , the translational slug velocity  $v_s$  and the volumetric phase ratio  $\phi = V_{\text{continuous}}/V_{\text{disperse}}$  between continuous and disperse phase, where  $V$  describes a volumetric flow rate [12, 13] go as far as to show that the use of a liquid–liquid slug flow capillary microreactor (SFCMR) enables control over conversion and selectivity in the solvent-free, phase-transfer catalysed epoxidation of methyl oleate (MO) to 9,10-Epoxy-stearic acid methyl ester (EAME), by adjusting these slug flow parameters. As shown in the reaction scheme in Fig. 2, hydrogen peroxide is used as a “green” oxidant [14]. The reaction is carried out using tungstophosphoric acid (TPA) as the oxidation catalyst and quaternary ammonium salt

**Fig. 1** Schematic representation of liquid–liquid capillary slug flow. Here  $L_s$  is the slug length,  $L_u$  is the distance between the front end of subsequent slugs,  $R$  is the radius to the wall film,  $R_0$  is the radius to the centre of the vortex,  $v_s$  is the translational slug velocity and  $h_{wff}$  is the height of the wall film



**Fig. 2** Reaction scheme of the solvent-free, phase-transfer catalyzed epoxidation of methyl oleate (MO) to 9,10-Epoxy-stearic acid methyl ester (EAME) using hydrogen peroxide with tungstophos-

phoric acid (TPA) as oxidation catalyst and quaternary ammonium salt Aliquat 336 (Q) as phase transfer catalyst [13]

Aliquat 336 (Q) as a phase transfer catalyst, enabling high yields without the need for a solvent [15].

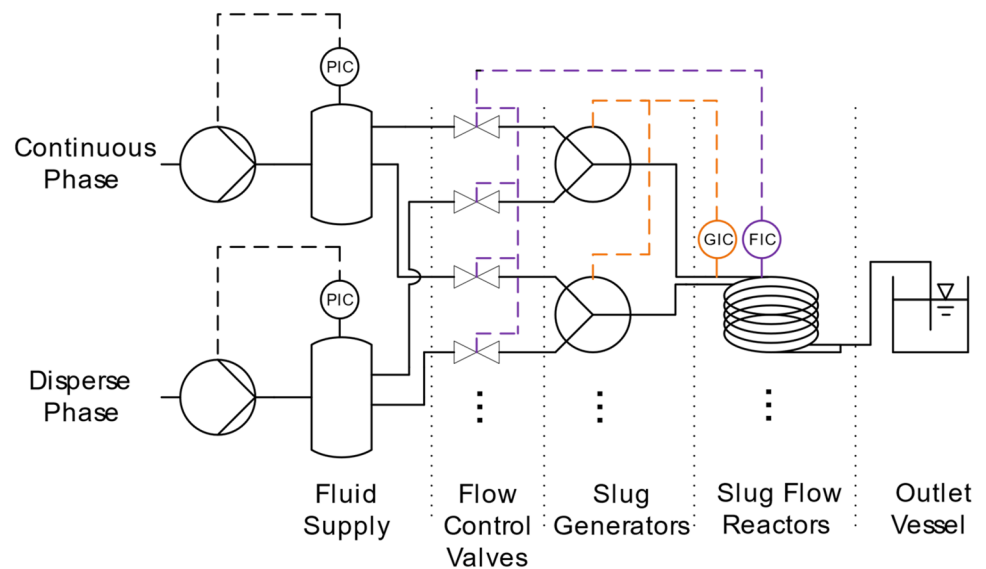
As the favourable effects associated with the liquid–liquid slug flow regime only appear in small capillaries, the scale-up of equipment like SFCMRs remains a challenge, with strongly limited throughput as the trade-off for intensified mass transfer [16]. Numbering-up by using parallel channels is one of the commonly employed methods of scale-up in capillary microreactors, and is the strategy used in this work. In the case of two-phase microreactors [17], differentiate between external and internal numbering-up, where the former describes the simple replication of entire devices to increase capacity, while the latter describes the replication of structures within a device as a means of scale-up. External numbering-up usually produces more predictable results but introduces redundancy and complexity. Internal numbering-up, on the other hand, aims to limit redundancy during numbering-up by replicating only (sub)structures with fixed dimensions and scaling all other structures and devices [18–20].

Prior research shows that for even fluid distribution in all parallel channels of a multi-capillary liquid–liquid contactor, some additional considerations must be made [17]: Although all parallel capillaries are nominally identical, manufacturing tolerances, corrosion, solid precipitation, or deposition are non-negligible for the fluid distribution across parallel capillaries at the micro scale. Therefore, to achieve uniform results across all units,  $v_s$ ,  $\phi$  and  $L_s$  must be measured and regulated in each capillary [21, 22]. For this purpose, Arsenjuk et al. [21] developed a concept for the active regulation of multichannel liquid–liquid contactors with the ability to control all slug flow parameters across several parallel capillaries. This is achieved by using pressure-controlled distributors, temperature-controlled

microchannels acting as rheological flow control valves (so-called thermorheological valves) and a type of adjustable slug flow generator capable of continuously altering slug length. A flowsheet of the concept is depicted in Fig. 3.

Wolffenbuttel [23] provides a summary of the four most common sensing techniques employed in liquid–liquid slug flow monitoring, including impedance-, absorbance- and reflectance sensors, as well as video imaging, which is concluded to be slow and elaborate. One commonly used approach involves using a video camera coupled with graph paper [23]. One approach involves tracking the slug flow using a high-speed camera, which provides high-resolution images and precise temporal information for analysing slug/bubble lengths, phase ratios, velocity, and bubble shape. Typically, the raw image data is converted into a binary grayscale representation to distinguish between different phases. By analysing the number of pixels, slug and bubble lengths can be deduced, and in conjunction with the snapshot and frame rate, the velocity can be calculated. While this method yields accurate results, the requirement for high-resolution images and high frame rates makes it unsuitable for industrial-scale numbering up tasks due to the associated cost of such a camera system [24, 25]. Another technique, microparticle image velocimetry ( $\mu$ PIV), can be utilized for flow characterization and visualizing internal slug flow patterns. However, like camera tracking,  $\mu$ PIV is also costly and adds complexity to data processing. These techniques are best suited for obtaining detailed insights into flow behaviour in laboratory-scale equipment and for calibrating other sensors [26]. Another method utilizes conductive fluids and measures impedance with electrodes, which is typically used for two-phase flow. While parameter determination is straightforward

**Fig. 3** Flowsheet of the parallel slug flow reactor control concept developed by Arsenjuk et al. [21]. PIC, GIC and FIC are the Pressure Controller, Slug Length Controller and Flow Controller respectively. Two parallel reactors are wrapped around each other and then placed into a water bath. Images of the original setup is included in Appendix C



with this method, the presence of electrodes can significantly impact the flow within the capillary [23]. Adding dye to one of the fluids offers a non-intrusive way to measure the parameters of interest by monitoring changes in absorbed light through the capillary. However, the use of additives may cause significant alterations in fluid properties [23]. A fourth method involves using modified optical fibre to discriminate between phases based on their reflectance. Nonetheless, the influence of the fibre on the capillary flow makes this method less desirable [23]. The final method utilizes two pairs of infrared emitters and detectors, with the signals processed through filters and a sampler. While this method allows monitoring of multiple channels in parallel, there is a trade-off between accuracy and computational expense [23]. In addition, Arsenjuk et al. [21] and Vietinghoff et al. [22] describe the use of infrared (IR) optical sensors. While the IR-based sensor may be realized in a non-intrusive and cost-effective way, at least one sensor per parallel capillary is required and the sensor is prone to irregularities in the slug flow pattern as well as requiring precise positioning. Moreover, adding reactants into the system, instead of inert substances, further increase the need for a sophisticated monitoring system to run the plant seamlessly and improve scalability. To address this, a video-based approach where raw video feed of the monitored capillaries is processed in real time by a computer vision algorithm to measure the slug length  $L_s$ , phase ratio  $\phi$ , and slug velocity  $v_s$  in multiple parallel capillaries is developed. By using a camera as the primary sensor equipment, multiple capillaries may be monitored using only one piece of hardware while the use of modern computer vision software allows for robust detection of the relevant slug flow features. In addition, the use of a camera and machine learning methods opens the possibility of future extensions to the image processing algorithm capable of making information usually gathered by visual inspection available to an automated control system.

The use of a camera is made possible by leveraging recent advances in neural-network-based object detection algorithms. Research by Joseph Redmon [27] and

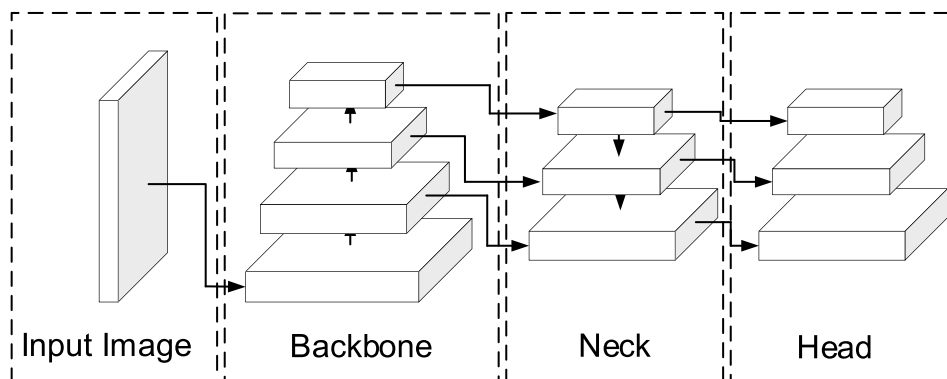
Bochovski et al. [28] gave rise to the YOLO-class of object detection algorithms. YOLO—an acronym for “you only look once”—is an object detector based on a convolutional neural network (CNN), capable of classifying and localizing objects within an image in a fast and efficient manner. Its speed is due to the single-pass working principle, where localization and classification is done on the entire image in one pass of the network. In contrast, algorithms like the R-CNN series, which employ a two-stage approach, first localize objects, and subsequently classify them in another algorithm sub-structure [29]. While the single-pass approach is slightly less accurate, it is efficient and capable of enabling real-time object detection on video feeds. As a CNN-based object detector, YOLO is also easily adaptable and may be retrained to detect many types of objects, provided labelled data sets exist [30].

The general structure of the YOLO-type object detection algorithm is shown in Fig. 4, where the input to the algorithm is the full image, represented as a tensor with three colour channels. The image is fed into the *backbone*, which consists of several convolutional layers responsible for feature extraction and down sampling of the image. The *neck* collects feature maps at different scales of the backbone and carries out pooling operations. Lastly, the *head* is responsible for predicting bounding box locations and objectness scores which represent the output of the algorithm. Due to the single-pass nature of YOLO, many overlapping bounding boxes are predicted for a singular object, making it necessary to cull bounding boxes by non-max suppression, taking into account the objectness scores and intersection-over-union values [28, 31].

As YOLO-type object detection algorithms only classify and locate objects in an image, further post processing of YOLO’s outputs is necessary to determine the desired slug flow parameters from images or video feeds. To achieve this, two different algorithms are developed in this work and compared in terms of performance and accuracy.

This paper focuses on sustainability and process intensification, by coupling the efficiency of the microreactor systems with the efficiency increase in deploying cutting

**Fig. 4** Generalized structure of a YOLO-type single-pass object detector [28]



edge technology for system automation by scaling up micro reactors for production level quantities, using sustainable green chemicals to produce in-demand products. A computer vision sensor for liquid–liquid capillary slug flow parameters is developed and applied in control of parallel capillary slug flow microreactors. These newly developed sensors enable observing multiple microreactors at once in real-time and the developed control system helps in using these high frequency control inputs to intelligently actuate the microreactor plant. The paper concludes with the proof of concept of a scaled-up (dual-channel) reaction system using the developed sensor and control system.

Section 2, outlines the experimental setup for data collection to train the YOLO-network. It also details the materials utilized for conducting these experiments, along with the adaptation steps required to repurpose the classic YOLO model for the proposed experimental system. Section 3 discusses the results of different iterations of the computer vision slug flow sensor are discussed. The evaluations primarily focus on determining the most efficient version, which is subsequently integrated into the dual-channel liquid–liquid SFCMR setup. To simulate practical industrial edge device usage, the system is executed on a mobile GPU.

## 2 Materials and methods

To develop the computer vision slug flow sensor, it was necessary to collect training data to retrain the YOLO object detector for the application in the system at hand. Furthermore, algorithms to extract and process the desired data from the output of the object detector were designed. To validate the resulting sensor as well as to test its performance, experimental setups were created. Specifically, one setup consists of a single capillary fed by syringe pumps to generate images for training and the determination of basic indicators of sensor performance. A second experimental setup was created to demonstrate

the ability of the sensor developed in this work to serve as part of a feedback control loop for a dual channel SFCMR carrying out the solvent-free epoxidation of methyl oleate in two parallel capillaries.

### 2.1 Experimental setups

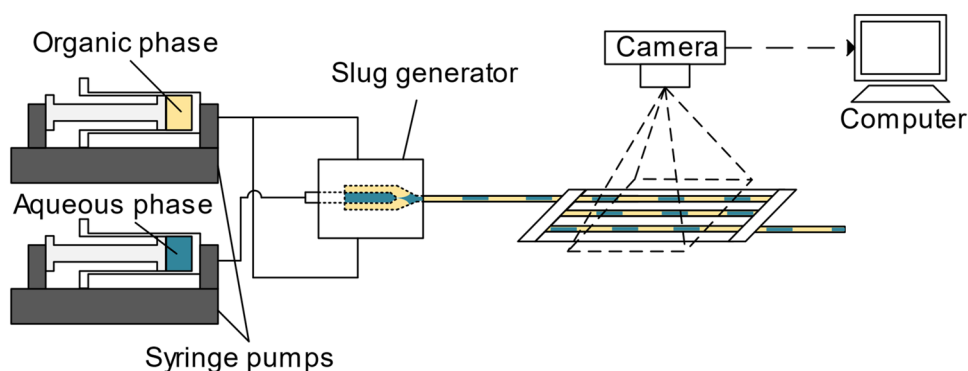
A two-phase system consisting of technical grade ca. 70 wt% methyl oleate (Sigma Aldrich, impurities consist of methyl stearate and methyl palmitate) as the continuous organic phase and demineralized water as the disperse (i.e., slug-forming) phase is used in sensor and controller development. In the test setup, syringe pumps send the two phases to a variable-geometry slug generator where the slug flow is formed [12]. By varying the geometry of the slug generator, the emerging slug length can be altered. The slugs flow through a flexible, transparent plastic capillary made of fluoroethylene propylene (FEP), with a nominal inner diameter of 1 mm. A section of the capillary tube is coiled and fixed on top of a white surface to simulate a multi-capillary setup. Above this the camera is positioned. The setup is illustrated in Fig. 5.

The camera setup consists of a vertical height-adjustable stand holding the camera (with specifications shown in Table 1) in place, in a location with minimum interference from external light sources. The video feed is transferred to a computer for processing via USB 3.0. The use of a single

**Table 1** Camera settings for training and validation dataset collection

Camera type	The imaging source DFK 37BUX265
Lens	Computar H0514MP2, 5 mm
Software	IC Capture 2.4
Resolution	1280 × 720
Camera distance from capillaries	50–250 mm
Max. Frames per second	20
Video codec and format	MJPEG/AVI

**Fig. 5** Flow diagram of the experimental setup used to calibrate the video sensor. A section of the capillary tube is coiled and fixed on top of a white surface to simulate a multi-capillary setup. The multi-capillary setup is shown in Fig. 6





capillary and syringe pumps allows for slug length  $L_s$ , phase ratio  $\phi$  and slug velocity  $v_s$  to be set and controlled precisely for sensor development and testing.

The data collected on the single-capillary test setup was then used to design the capillary reactor test setup. A flow diagram of the dual-channel liquid–liquid SFCMR is shown in Fig. 6. It consists of two pressurized stainless-steel holding vessels B1 and B2, thermorheological flow control valves TRV 1–4, automatically adjustable slug generators SG 1 and 2, phase separator B4, and automatically adjustable control valves VC8, VC11 and VC12 (Swagelok SS-SS1-A). Here, the computer vision sensor fulfils the measurement functionality of slug length controllers GIC 1 and 2, phase ratio controllers QFI(C) 1 and 2, as well as slug velocity controllers SI(C) 1 and 2. The reaction zones (REAC) consist of FEP capillary with a length of 15 m each, suspended in a water bath at 60 °C. The reactants are pressurized to 0.5 MPa in the holding vessels for propulsion to the slug generators.

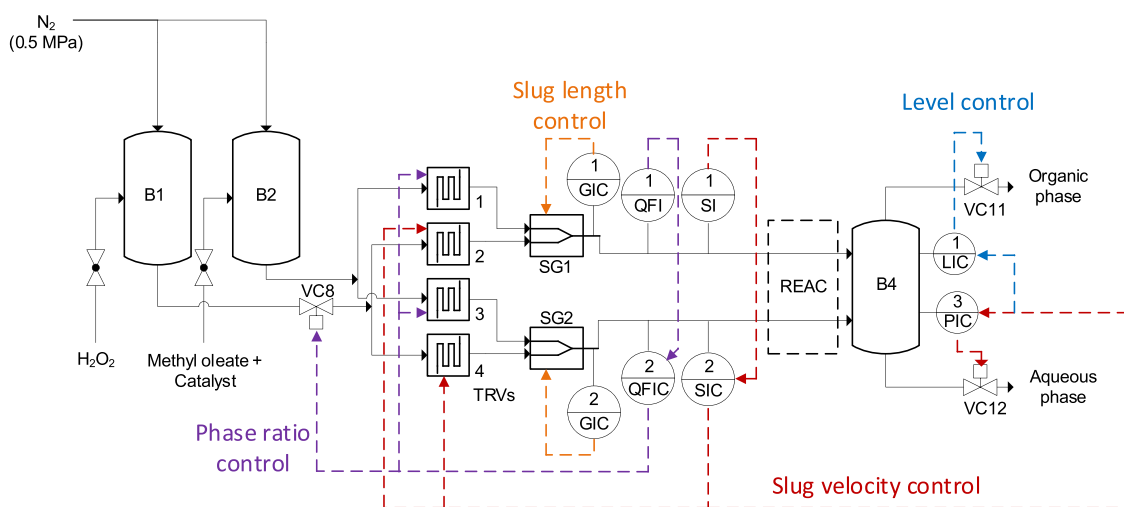
The control targets are: (1) Both channels should achieve and maintain a set average parameters  $\bar{v}_s$ ,  $\bar{\phi}$ , and  $\bar{L}_s$  and (2) parameters  $v_{s,i}$ ,  $\phi_i$ , and  $L_{s,i}$  should remain similar in both channels during normal operation to ensure predictable results during numbering-up. Because both reactors are connected both supply-side as well as downstream, they act as interconnected channels. As a practical means of addressing the coupling behaviour brought on by this, control of  $\phi_i$  and  $v_{s,i}$  is decomposed into control of the average across all reactor channels, as well as control

of the difference between both channels using the “Delta-Controller” developed for this task. Further details on this can be found in Appendix A. Slug length  $L_s$  is controlled by setting the slug generator insertion depth. This is done on a per-channel basis as the adjustment of slug length shows little coupling with the other parameters. There is sufficient dead time in the micro reactor after the slug generator and before the reaction zone (REAC), where the cool reactants form stable slugs and the camera sensor is used to measure slug length, phase ratio and slug velocity. This helps in isolating the actual reaction zone, which could be exposed to higher temperatures without affecting the measurements upstream.

## 2.2 Epoxidation catalyst

As previously mentioned, the solvent-free epoxidation of methyl oleate using hydrogen peroxide requires a catalyst to achieve meaningful conversion. Contrary to previous works, here the active species consisting of TPA ( $H_3[PW_{12}O_{40}]$ ) and Q ( $Ni[(CH_3)(C_8H_{17})_3]Cl$ ) is not formed in-situ, but rather pre-synthesized [13]. This is done as the component TPA is present as a solid at room temperature and promotes the decomposition of the 35%-w  $H_2O_2$  used to conduct the reaction. Secondly, TPA has been found to cause deposits within the TRVs, especially during cooling.

Because full or partial blockages are detrimental to the performance of micro reactors in general, a different approach is necessary. Therefore, the active species—also



**Fig. 6** Flow diagram of the dual-channel SFCMR setup including main control loops, pressurized holding vessels B1 and B2, thermorheological valves TRV 1–4, slug generators SG 1 and 2, reaction zones (REAC) with a length of 15 m each, phase separator B4, and automatically adjustable control valves VC8, VC11 and VC12. GIC

1–2 represent the Slug Length indicators and controllers, QFI and QFIC represent the Phase Ratio indicator and controller respectively, SI and SIC represent the Slug Velocity indicator and controller respectively, LIC represents the phase separator level indicator and controller and PIC represents the Pressure indicator and controller

referred to as *Venturello Complex* ( $Q_3[PW_{12}O_{40}(O)]$ )—which presents as an ionic liquid soluble in methyl oleate, is prepared according to a modified procedure adapted from kinetic studies by Maiti et al. [32], as well as Wang and Huang [33]. For a detailed description of the preparation procedure used in this work refer to Appendix B.

### 2.3 Computer vision slug flow sensor

The three parameters of interest—slug length  $L_s$ , volumetric phase ratio  $\phi$ , and slug velocity  $v_s$ —can be divided into two groups: Spatial and temporal. In principle, determining  $L_s$  and  $\phi$  requires spatial detection and classification in single video frames, while  $v_s$  estimation requires extraction of temporal information on the detections in sequential video frames. As shown in Fig. 7, raw image data is first captured and pre-processed, which includes cropping and resizing of the frame, as well as converting the image data into a suitable array format. Next, the object detection algorithm determines spatial parameters, which are also stored in internal memory. Finally, using current and previous spatial data, temporal processing takes place to yield the corresponding parameter.

The algorithm is implemented using *Python* with its packages *Keras* and *TensorFlow* for machine-learning tasks, *OpenCV* for image processing and *NumPy* for numerical operations. As the use of graphics processing units (GPU) provides speed advantages over typical central processing units (CPU) in the calculations associated with artificial neural networks, GPUs are employed in this work [34]. Specifically, the *Nvidia GeForce* GPUs *RTX2070* (training and inference), and *940M* (inference) are used to provide benchmarks at different levels of hardware capability and cost.

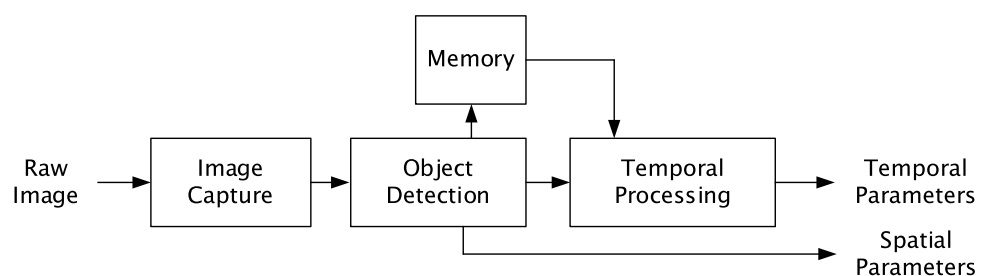
#### 2.3.1 Spatial parameter estimation using object detection

As convolutional neural networks (CNN) have found widespread application in computer vision tasks involving object detection—often vastly outperforming prior methods in terms of accuracy and speed—the use of a CNN-based algorithm for the extraction of information on spatial slug flow parameters from a video feed appears

reasonable [35, 36]. For this, two possible solutions can be considered: In the first, a pre-existing object detector network is trained on hand-labelled images of slugs in capillaries, and the two parameters are calculated using elementary geometry on the detection coordinates. Alternatively, a custom network that directly estimates the two parameters from a raw video feed could be developed. While the latter may result in a fast, purpose-built solution, it would likely be very time-consuming. Therefore, the large volume of available research is leveraged by using the existing YOLO object detection algorithm at the core of the computer vision slug flow sensor. In addition to being fast and simple to re-train for the detection of slugs, the YOLO algorithm holds two major benefits for the use in a slug flow sensor: Its outputs are in the form of rectangular bounding boxes around detected objects, which conveniently approximate the shape of a slug, and detection of multiple objects per frame is inherently supported, allowing the monitoring of multiple channels with minor additional post-processing [28].

In the context of our study, we have made a deliberate decision to focus on using bounding boxes for slug shape representation, rather than exploring alternative techniques such as contour-based segmentation methods or higher-order geometric models. Our choice is based on considering the specific objectives and requirements of our study. By utilizing bounding boxes, we have achieved satisfactory results in detecting and characterizing slugs accurately for the purpose of our proposed detector. This approach aligns with the simplicity, efficiency, and practicality that are essential for our research goals. As the YOLO-algorithm has received much attention in recent years, several versions with slightly different architectures have been developed by the original authors and others. Therefore, the versions YOLOv3 and YOLOv4, as well as the reduced model YOLOv4-tiny, are investigated for the use in the computer vision slug flow sensor. Other models considered have been discussed in Appendix D, however only the best models are chosen for comparison purposes. YOLOv3 employs the full *Darknet53* backbone, a feature pyramid network as the neck and three YOLOv3 heads, while YOLOv4 uses the *CSPDarknet53* backbone, including several optimizations over *Darknet53* which make it more accurate but also slower in certain instances, as

**Fig. 7** Schematic representation of the image processing algorithm to extract slug flow parameters from a sequence of raw images



well as pooling layers and a path aggregation network as the neck. [28, 37]. The heads of YOLOv4 remain unchanged from previous iterations. YOLOv4-tiny uses the smaller *CSPDarknet53-tiny* backbone with a feature pyramid network neck and two YOLOv3 heads, instead of three [38]. All versions of YOLO are implemented using the *TensorFlow* machine learning framework. As inputs, square images with a width  $\sigma$  of 416 pixels and 608 pixels are tested, and one class (slug) is defined for classification, resulting in the models YOLOv3-608, YOLOv4-608, YOLOv4-416 and YOLOv4-tiny-416, where the number refers to the width of the image the model accepts as its input.

The dataset used for training, validation and testing consists of 340 unique images at different slug lengths  $L_s$  (1.8–8.3 mm), as well as varying slug velocities  $v_s$  (0–26 mm  $s^{-1}$ ). All images were captured at a camera height of 120 mm. To further improve the robustness of the training result, the dataset is augmented by flipping images horizontally and vertically, as well as adjusting brightness up and down by up to 40%, resulting in a set of 819 images. The set is split into a training set (80%), validation set (10%), and testing set (10%).

The output of the object detector is a matrix of normalized bounding box coordinates with the structure illustrated in Fig. 8 below. The bounding boxes determined by the object detector approximate the slugs as rectangles. As the detections appear in the coordinate matrix in no order, an algorithm is introduced which dynamically assigns channel coordinates according to the y-components of the detection matrix and sorts the detections into their respective channels on each frame.

The vector of estimated slug lengths  $\vec{L}_s = (L_{s,1}, \dots, L_{s,n_c})$  is determined from the coordinate matrix  $C$  shown in Fig. 8. Here, each element of  $\vec{L}_s$  denotes the maximum estimated slug length of a respective channel within a frame, while  $n_c$  is the number of detected channels in the frame. With the horizontal coordinates  $x_{i,1}$ , and  $x_{i,2}$  from matrix  $C$ , as well as the pixel density  $d_p$  (in pixels per millimeter), the length of a slug  $L_{s,i}$  for a row of matrix  $C$  can be determined as follows:

$$(x_{i,2} - x_{i,1}) \cdot d_p = L_{s,i} \tag{1}$$

As a single channel may have multiple slugs in a single frame, a means of reducing the detected slug lengths to one slug per channel is necessary. For the larger model YOLOv4-416, which could be trained to exclude partial slugs at the edge of the frame, applying the channel-wise average appeared suitable. The smaller model YOLOv4-416-tiny, on the other hand, could not be trained the exclude partial slugs, therefore the channel-wise maximum was used to return sensible slug length estimations.

In addition to the slug length measurements, the detection coordinate matrix  $C$  and channel coordinates are used to create a binary sequence array  $S$ , which is used in the channel-wise estimation of volumetric phase ratios  $\varphi_k$ .

$$S = \begin{bmatrix} s_{1,1} & \dots & s_{1,\sigma} \\ \vdots & \ddots & \vdots \\ s_{n_c,1} & \dots & s_{n_c,\sigma} \end{bmatrix}, s_{k,m} \in (0, 1) \tag{2}$$

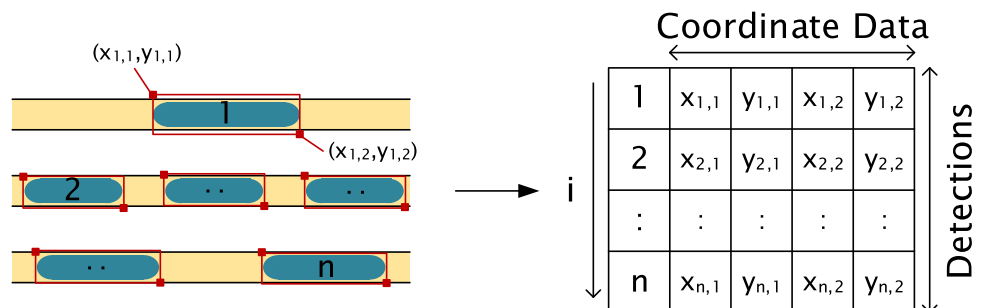
This is achieved by initializing an array of binary values the size of the input image with zero. Using the detection coordinates, the initialized array is augmented by writing ones at the indices enclosed by the bounding boxes creating the binary stencil array  $A$ . Lastly, the array is simplified by including only the rows at the channel coordinates, yielding the binary sequence array  $S$  of size  $n_c \times \sigma$ . A graphical representation of this algorithm is shown in Fig. 9.

From the sequence array  $S$ , the vector of estimated volumetric phase ratios  $\vec{\varphi} = (\varphi_1, \dots, \varphi_n)$  can be calculated according to Eq. 3, where  $J$  is a  $\sigma \times n_c$  matrix of ones and  $\oslash$  denotes the element-wise, or *Hadamard*, division between the resulting vectors.

$$\vec{\varphi} = [SJ] \oslash [(J^T - S)J] \tag{3}$$

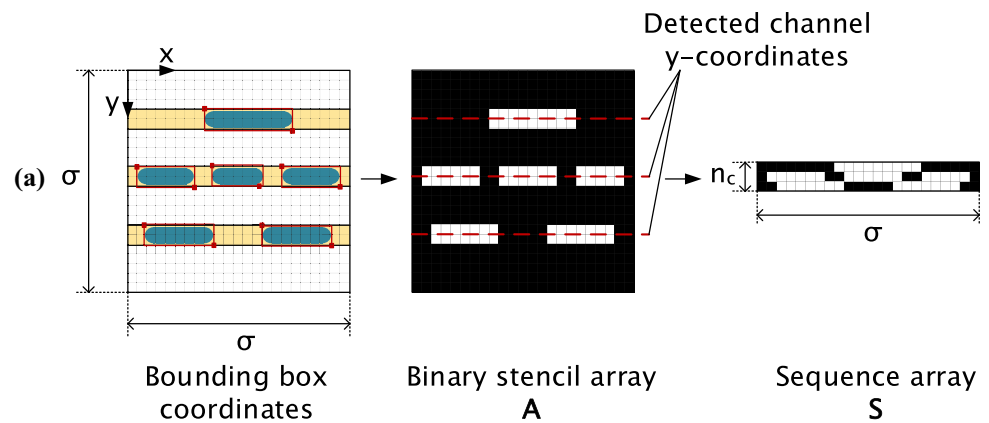
For the validity of Eq. 2, it is assumed that slug detections which include the entire width of the frame do not occur. As slugs of this length lie outside the intended working range of the sensor, this limitation is acceptable.

**Fig. 8** On the left, bounding boxes (red) approximate the shape of slugs (blue) within an image. On the right, the corresponding output of the object detector in the form of a coordinate matrix  $C$





**Fig. 9** **a** Algorithm for the creation of the binary sequence array  $S$ . The bounding box coordinates are used to create the binary stencil array  $A$ , which is reduced to include only the rows at the detected channel  $y$ -coordinates  $\hat{y}_c$ . This yields the binary sequence array  $S$ . **b** A single row of a sequence array  $S$  next to the image it was generated from



### 2.3.2 Slug velocity estimation by temporal processing of spatial slug flow measurements

Estimation of the vector of slug velocities  $\vec{v}_s = (v_{s,1}, \dots, v_{s,n_c})$  is possible by considering consecutive images from a video feed and evaluating the movement of the detected slugs for each respective channel. For this, two approaches are developed and investigated: In the first, the YOLO object detector is extended by an additional neural network structure which accepts a series of the sequence arrays  $S(t)$  of consecutive images from a video feed as its input and returns the vector of estimated slug velocities  $\vec{v}_s = (v_{s,1}, \dots, v_{s,n_c})$  as its output. In the second approach, consecutive sequence arrays  $S(t)$  are processed using an explicit calculation scheme, based on matrix–vector operations.

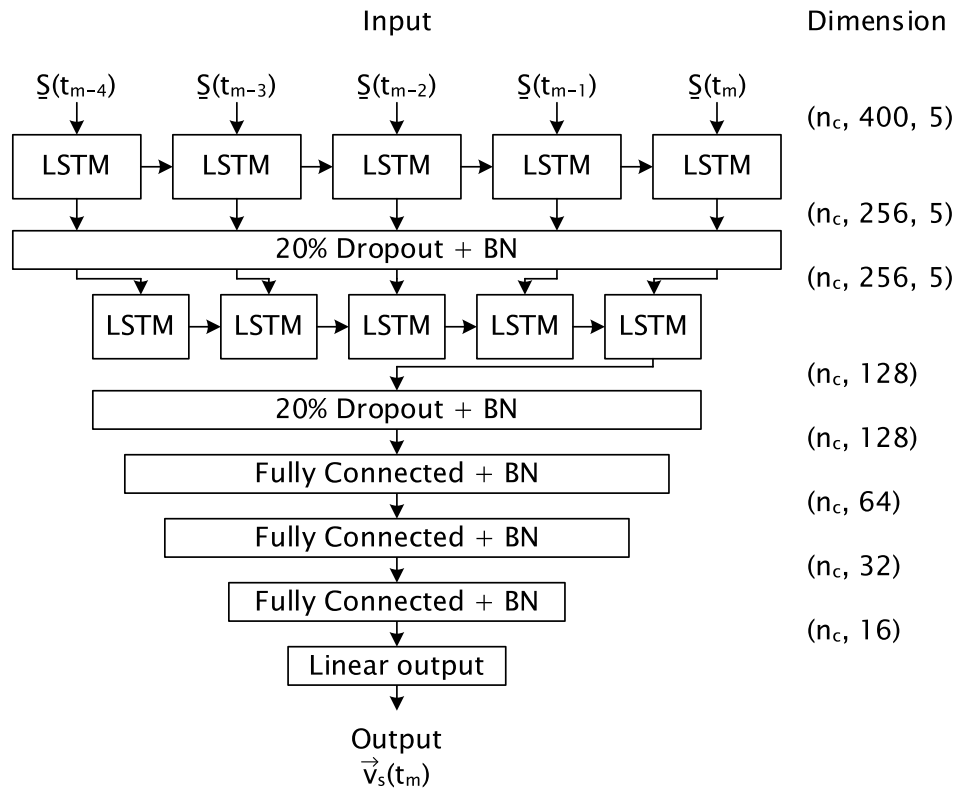
The neural network approach is realized using a long short-term memory network (LSTM) to create a robust algorithm for accurate and precise slug velocity estimation. LSTM networks are a widely used variant of recurrent neural networks (RNN) introduced by Hochreiter and Schmidhuber [39], which tackle important shortcomings of recurrent networks developed previously. During the training of RNN, error signals flowing through the feedback loop tend to either (a) blow up or (b) vanish since the evolution of the backpropagated error exponentially depends on the size of the weights. Case (a) may lead to oscillating weights, while in case (b) learning to bridge long time lags can take prohibitively long or does not work at all. The originally proposed architecture solves

this by enforcing constant error flow-through internal states of specific units and truncated gradients at certain points [39]. There are several LSTM variants found in the literature, of which the one introduced by Gers et al. [40] will be outlined here, as this version is implemented in the *Keras* API. The central idea behind the LSTM architecture is a memory cell that can maintain its state over time, and non-linear gating units that regulate the information flow into and out of the cell [41]. In addition to the input and output gates, the cell described by Gers et al. [40] also uses a forget gate which can decide what information will be discarded from the cell state. This modification is introduced as the authors observed that the cell states often tend to grow linearly during the presentation of a time series, which leads to the saturation of the output activation function and thus vanishing of the gradients during training.

Several iterations of the LSTM slug velocity estimator were considered, with the best results achieved by the model 5-LSTM(256-128)-FC(64-32-16). It accepts an array of five  $n_c \times 400$  centre subsets  $\underline{S}$  from a moving window of consecutive sequence arrays  $S$ , beginning at the time of measurement  $t_m$ , as its input. Each of the five subsets  $\underline{S}$  is fed into an LSTM block with batch normalization and 20% dropout. After this, alternating fully connected layers and batch normalization are used. This structure is shown in Fig. 10. More information about the model candidates and the selection criteria can be found in Appendix E.

The explicit calculation scheme, on the other hand, is meant to be a simple and fast approach to the problem

**Fig. 10** Structure of the LSTM slug velocity detector 5-LSTM(256-128)-FC(64-32-16)



of slug velocity estimation. For this, two core operations are necessary: First, the number of detected slugs in each channel is determined from the sequence array  $S(t_m)$  at the time of measurement  $t_m$ , which is subsetting into its left  $\sigma - 1$  columns  $\bar{S}_1$  and its right  $\sigma - 1$  columns  $\bar{S}_2$ . By subtracting the two submatrices and taking the absolute value of each element, the edges of the detected slugs are highlighted in the resulting matrix  $B$ . The row-wise sum, here achieved by multiplying  $B$  with  $(\sigma - 1) \times n_c$  matrix of ones  $J_n$ , yields the vector  $\vec{n}_t$  of total transitions in each channel:

$$\bar{S}_1 = \begin{bmatrix} s_{1,1} & \dots & s_{1,\sigma-1} \\ \vdots & \ddots & \vdots \\ s_{n_c,1} & \dots & s_{n_c,\sigma-1} \end{bmatrix}, \bar{S}_2 = \begin{bmatrix} s_{1,2} & \dots & s_{1,\sigma} \\ \vdots & \ddots & \vdots \\ s_{n_c,2} & \dots & s_{n_c,\sigma} \end{bmatrix} \quad (4)$$

$$B = \begin{bmatrix} |s_{1,1} - s_{1,2}| & \dots & |s_{1,\sigma-1} - s_{1,\sigma}| \\ \vdots & \ddots & \vdots \\ |s_{n_c,1} - s_{n_c,2}| & \dots & |s_{n_c,\sigma-1} - s_{n_c,\sigma}| \end{bmatrix} \quad (5)$$

$$\vec{n}_t = \mathbf{B} \mathbf{J}_n \quad (6)$$

With this, the vector of estimated slug velocities  $\vec{v}_s$  is determined by subtracting the two consecutive sequence arrays  $S(t_{m-1})$  and  $S(t_m)$ , taking the element-wise absolute value to yield matrix  $S_\Delta(t_m)$ , and finally applying the

row-wise sum as well as scaling factor  $f$ , dependent on frame-time  $t_f$  and pixel density  $d_p$ :

$$S_\Delta(t_m) = \begin{bmatrix} |s_{1,1}(t_{m-1}) - s_{1,1}(t_m)| & \dots & |s_{1,\sigma}(t_{m-1}) - s_{1,\sigma}(t_m)| \\ \vdots & \ddots & \vdots \\ |s_{n_c,1}(t_{m-1}) - s_{n_c,1}(t_m)| & \dots & |s_{n_c,\sigma}(t_{m-1}) - s_{n_c,\sigma}(t_m)| \end{bmatrix} \quad (7)$$

$$f = \frac{1}{t_f d_p} \quad (8)$$

$$\vec{v}_s(t_m) = f \{ [S_\Delta(t_m) \mathbf{J}] \odot \vec{n}_t \} \quad (9)$$

As Eq. 9 is only valid so long as no element of  $\vec{n}_t$  is zero, safeguards must be in place. Because channel detection is performed on every frame, in the absence of slugs in one channel, the channel is simply not detected as such, and no estimated slug velocity is calculated.

### 3 Results

In the following section, the different versions of the computer vision slug flow sensor developed in this work are evaluated. Furthermore, the performance of the most

efficient version of the sensor developed in this work, YOLOv4-416-tiny with explicit temporal processing, is implemented as part of the dual-channel liquid–liquid SFCMR setup and executed using the GeForce 940M mobile GPU to simulate the use of edge devices available for industrial settings.

### 3.1 Performance of the computer vision slug flow sensor

Important performance indicators of the computer vision slug flow sensor are its accuracy in estimating spatial and

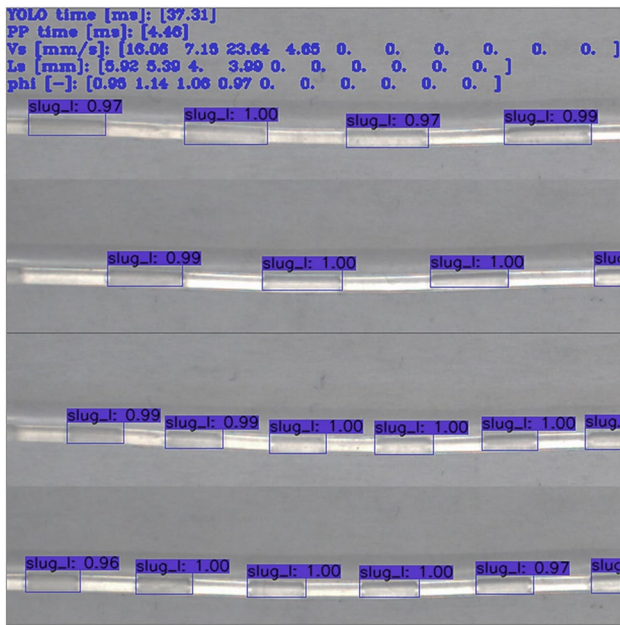
temporal slug flow parameters, as well as the maximum possible sampling rate on a video stream. Figure 11 shows a graphical user interface, designed to observe the computer vision slug flow sensor during operation. In this case, four parallel channels are monitored simultaneously. The indicators “YOLO time” and “PP time” show the execution time for the YOLO object detector and the post-processing including channel detection, spatial, and temporal processing, respectively.

Owing to the way the YOLO class of models infer from images, it is expected that increasing the number of objects in an image does not increase the inference time [31]. This is confirmed by the data presented in Fig. 12, as execution time barely increases with the number of channels as in the case of the model YOLOv4-416.

Execution time of the object detector is instead dictated primarily by the version of YOLO used as well as the input size of the object detector. This is shown in Fig. 13, where execution time decreases as the size of the input frame decreases. It is also apparent that for this task, YOLOv4 holds a clear speed advantage over the older YOLOv3.

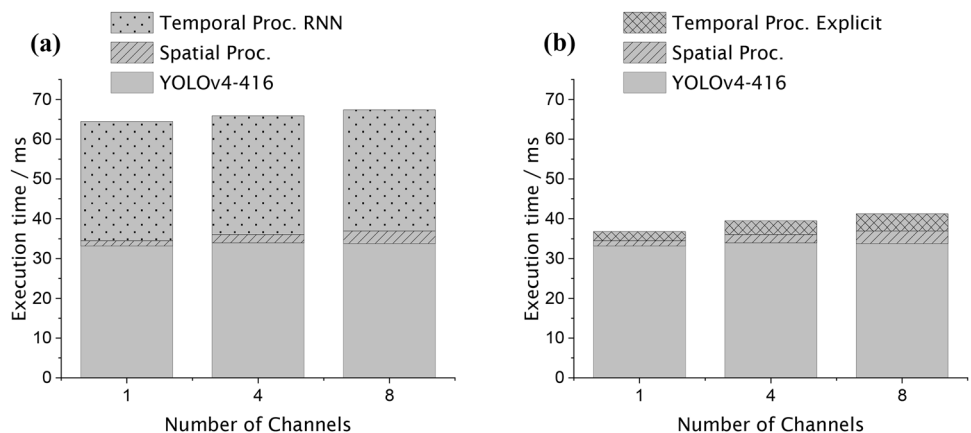
The computer vision slug flow sensor’s accuracy can be ascertained from the parity plots in Fig. 14: Plots (a) and (b) compare average estimates by the computer vision slug flow sensor with average manual measurements of five-second video segments. Measurements are taken at a constant phase ratio  $\phi$  of 1. For comparison purposes, results are shown for temporal parameter estimation by both RNN and explicit calculation scheme, as well as spatial parameter estimation by YOLOv4-416 and YOLOv4-416-tiny.

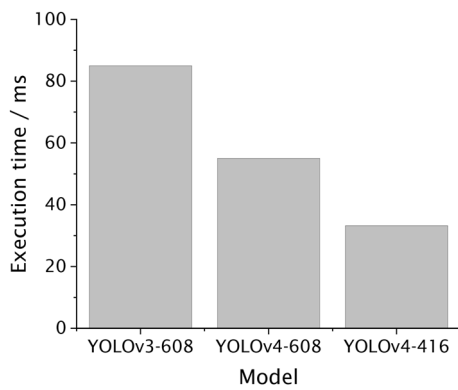
The mean absolute error of slug length estimation is 0.22 mm for YOLOv4-416-tiny and 0.15 mm for YOLOv4-416, resulting in mean relative errors of 3.41% and 3.30% for slug length estimation across all tested settings, respectively. The maximum relative error for slug length estimation is no more than 7.0% for both models, with the highest relative errors occurring at the edge of the tested range of slug lengths. Similarly, the mean absolute errors



**Fig. 11** Slug detection on four channels. YOLO Time, PP time, slug length  $L_s$ , slug velocity  $v_s$  and volumetric phase ratio  $\phi$  and their respective units have been shown at the top left corner for each individual slug in an array form. The system was also checked for multiple capillary processing as can be seen in Appendix F

**Fig. 12** Execution times of the individual sections of the algorithm (object detection, spatial processing, and temporal processing), using the YOLOv4-416 object detection model, using **a** RNN-based, and **b** explicit temporal processing. Hardware: CPU—Intel i7-6500U, GPU—Nvidia RTX 2070





**Fig. 13** Execution time of inference on a single capillary using different versions and input sizes of YOLO. Hardware: CPU—Intel i7-6500U, GPU—Nvidia RTX 2070

of slug velocity estimation are  $0.35 \text{ mm s}^{-1}$  and  $0.47 \text{ mm s}^{-1}$  for explicit calculation scheme and RNN, respectively. The mean relative errors are 2.5% for the explicit calculation scheme and 3.5% for the RNN. The maximum relative error is no more than 7.3% at the low end of the tested range of slug velocities.

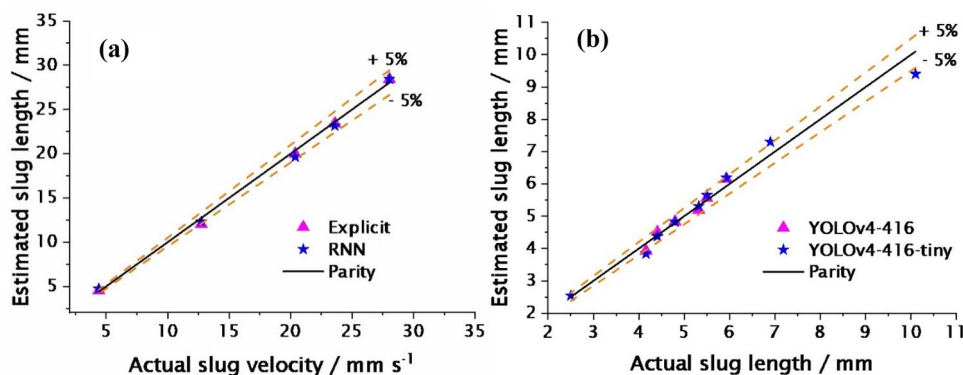
As YOLO is a data-based model, its accuracy of object detection is limited by the labelled data used for training. Furthermore, as indicated by parity plot Fig. 14b, the estimation accuracy appears to be less accurate at slug lengths which were not included in the training set (1.8 mm, 2.2 mm, 3.0 mm, 6.0 mm, and 8.3 mm), resulting in an underestimation of slug length at the high end of the scale.

Figure 15 shows the results of a dynamic test run at a constant phase ratio of 1. Slug length is manually decreased in frequent, small steps over two minutes to avoid severely irregular flow patterns during the

transience. The inverse response of slug length to the adjustment of slug generator geometry, as described previously by Arsenjuk et al. [12], can be observed at each step. The detected slug velocity values exhibit some variation, showing peaks at points where the slug length is adjusted. This slight dependency of velocity on the slug length is expected due to the characteristics of the slug generator, which undergoes a change of its internal volume at the time of slug length adjustment. As the slug length curves show, both YOLOv4-416 as well as YOLOv4-416-tiny perform similarly, however the full model tends to show deeper troughs and slightly slower response to slug length changes. This is likely less due to the different intrinsic capabilities of the two models. Instead, this effect is caused by the different post-processing for each model. Taking the channel-wise average, as with YOLOv4-416, leads to a delay in response time but returns more precise estimates, while applying a channel-wise maximum leads to a faster response time, especially in situations where the slug length is increasing, but also the over-estimation of slug lengths as long as slugs longer than the most recent one is visible within the frame.

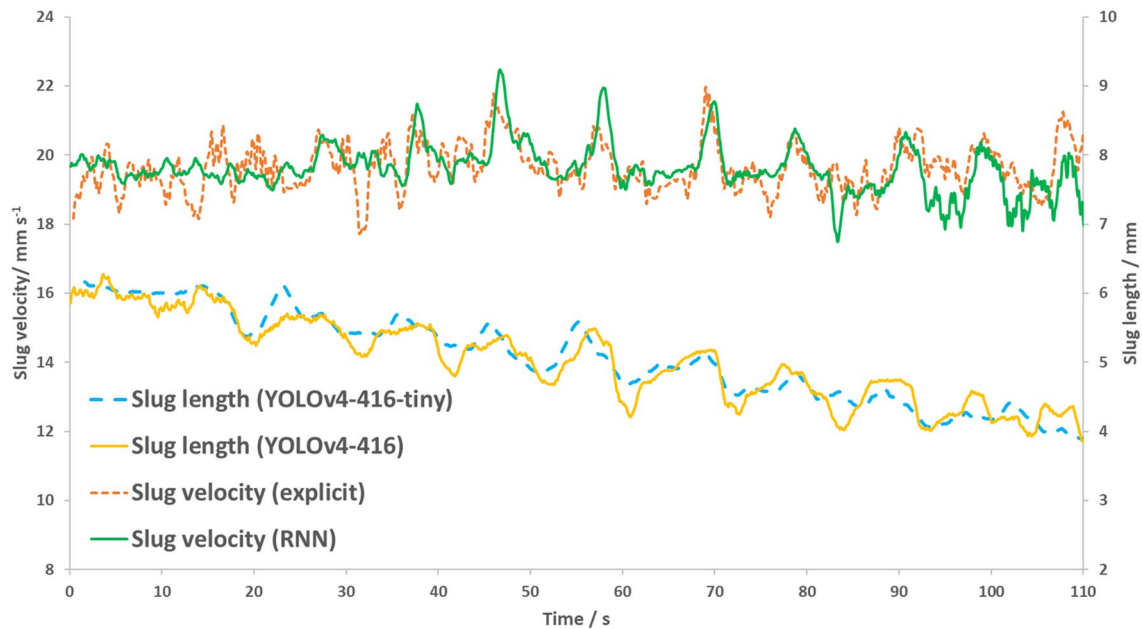
Figure 16 shows the results of second dynamic test run where the slug velocity is changed at a constant slug length setting and a phase ratio of 1. A notable detail in this run is the non-dependence of slug length on slug velocity.

It is visible that both the explicit, as well as the RNN-based algorithm, show very similar dynamics. The estimates provided by the RNN-based algorithm appear to be slightly smoother than those returned by the explicit temporal processing scheme. This is likely caused by the fact that the RNN-based approach uses samples from five consecutive points in time, making the RNN-based algorithm exhibit properties of a filter. A similar reduction of

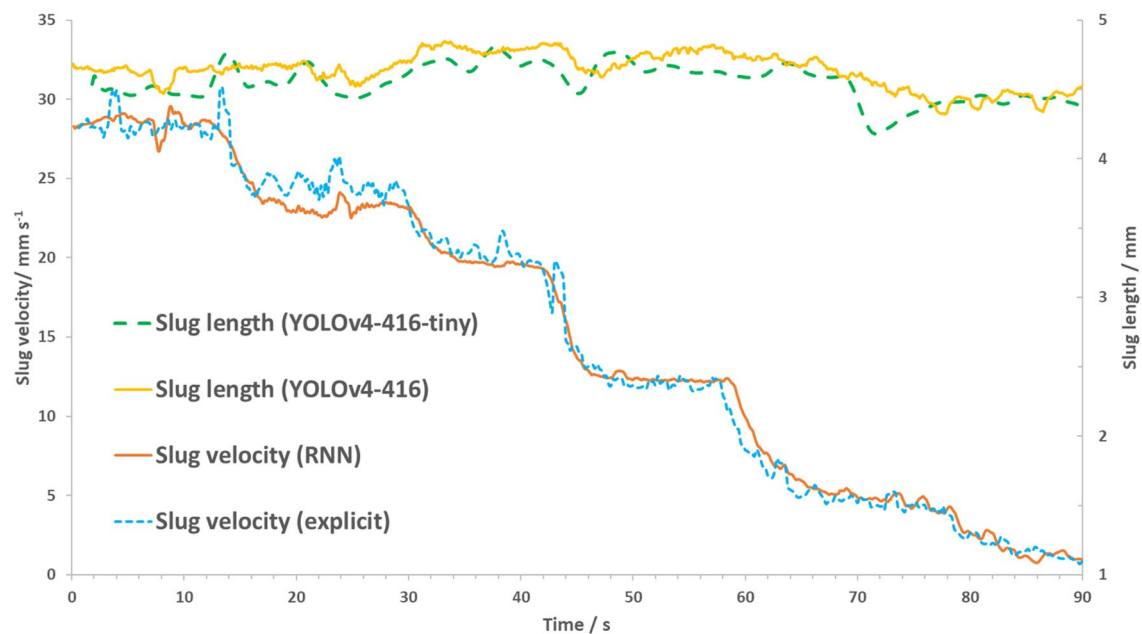


**Fig. 14** Parity plots for slug velocity  $v_s$  (a) and slug length  $L_s$  (b). The plots compare mean estimates by the computer vision slug flow sensor with mean manual measurements of five-second video segments of a single capillary. Measurements are taken at a constant

phase ratio  $\phi$  of 1. For comparison purposes, results are shown for temporal parameter estimation by both RNN and explicit calculation scheme, as well as spatial parameter estimation by YOLOv4-416 and YOLOv4-416-tiny



**Fig. 15** Dynamic test run with decreasing slug length at a constant velocity—unfiltered data. Phase ratio  $\phi = 1$



**Fig. 16** Dynamic test run with decreasing velocity at unchanged slug generator setting—unfiltered data. Phase ratio  $\phi = 1$

variability in the results of the explicit temporal processing scheme can be achieved by employing a simple moving average filter at the expense of a fast response time.

All development and testing discussed above was conducted with cropped, but otherwise unadulterated images and video feed at native camera resolution and a vertical camera height of 120 mm. Considering the goal

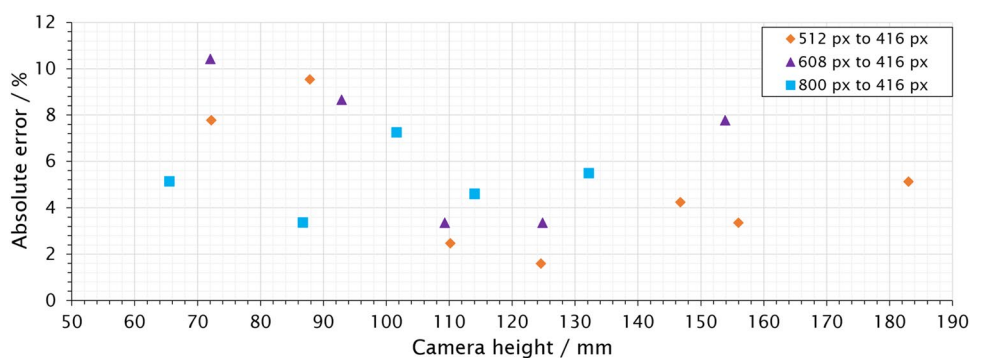
of developing a computer vision slug flow sensor suitable for the fast and flexible parallelization of capillaries (e.g., in capillary reactors), the degree to which the sensor algorithm can function at different camera heights as well as with scaled video feed without re-training was investigated. As the number of capillaries visible in each frame is dependent on the distance of the camera from



the capillaries, the robustness of the developed sensor algorithm is tested using camera distances between 50 and 200 mm. Additionally, because the distance of the camera from the capillaries cannot be increased arbitrarily without compromising clarity, higher resolution images are captured and scaled to the input size of the respective object detection models. The results are shown in Fig. 17:

The error is lowest at camera heights closest to 120 mm—the height used to record the training data. Consequently, YOLOv4-416 performs less accurately at camera heights deviating from the camera height used for the training data, however, even at camera heights of up to 183 mm, the average absolute error could be acceptable for many applications at 5.1% while yielding a 52.5% increase in view window width available for additional capillaries. Furthermore, the average absolute error rises quickly with a decrease in camera height despite expectation that a more detailed image should lead to a more accurate estimate. As the YOLO-class of models has no prior “knowledge” of slugs, this would be a misconception. Additionally, the larger view window may be beneficial to estimation accuracy in the sense that it offers a larger number of slugs for averaging, making the computer vision slug flow sensor relatively less sensitive to disturbances and failed detections. Similarly, capturing video using larger resolutions and scaling it to fit the model input size results in less accurate results, the further the resolution used deviates from the resolution used for training. A notable detail is the relatively high accuracy of YOLOv4-416 at a camera height of 86 mm and a capture resolution of  $800 \times 800$  px. This is likely caused by the opposing effects of scaling the image down, which results in reduction in size of the slugs in the view window and the reduction in camera height, which increases the size of the slug, indicating that the size of the slug within the image relative to the total size of the image is a critical variable for training and should be considered in the selection of training data sets for future iterations of the sensor.

**Fig. 17** Average absolute errors observed in slug length estimation (actual slug length of 5.6 mm) using YOLOv4-416 with video feed of sizes  $512 \times 512$  px,  $608 \times 608$  px, and  $800 \times 800$  px scaled to  $416 \times 416$  px at different camera heights between 50 and 200 mm



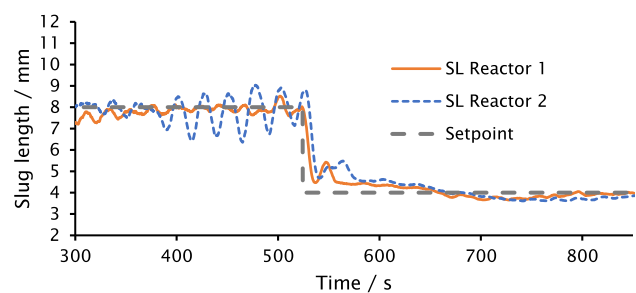
## 3.2 Sensor performance in feedback control loops

To evaluate the performance of the sensor as part of a feedback control system and present a low-redundancy control concept for such reactor setups, it is implemented in the two-capillary liquid–liquid SFCMR carrying out the solvent-free epoxidation of methyl oleate to 9,10-epoxystearic acid methyl ester presented in Sect. 2.1.

### 3.2.1 Slug length control

Because the slug lengths only have a minor impact on the other slug flow parameters and vice versa, their control is rather uncomplicated in terms of inter- and intra-capillary coupling and can be achieved using separate SISO PI-controllers. Figure 18 shows a simultaneous step change of both reactors' slug length setpoints:

After the step change, the system takes about 60 s to bring the slug lengths of both reactors to within a 10% margin of the setpoint and about 100 s until fully settled. A notable challenge in slug length control lies in the fact that long slug lengths over five millimeters effectively produce fewer measurement updates due to the fixed size of the measurement window, leading to lag effects and an emphasis on the natural variation of the slugs produced by the slug generators. Furthermore, the system response to a change in slug generator position elicits an inverse response. In combination, this leads to oscillations as seen between 380 and 500 s. These oscillations may be reduced



**Fig. 18** Slug length setpoint change

by the reduction of the PI-controller gain at the cost of a slower step response and a longer settling time. Ultimately, slug lengths greater than six millimeters are not particularly relevant for the task at hand (the epoxidation of methyl oleate) and thus present an extreme case, but in future iterations, a controller gain schedule could help mitigate the effect of these nonlinearities.

### 3.2.2 Slug velocity and phase ratio control

A set of step inputs to the phase ratio setpoint is depicted in Fig. 19. Between zero and 600 s, the setpoint tracking for both reactors is satisfactory, keeping within a 10% margin of the setpoint for 91.8% of the time. At a phase ratio of 6 and after step inputs to the phase ratio setpoint, the process shows more variability, mostly in the form of oscillations about a mean. This behavior settles once the “Delta”-controller becomes active, which is indicated by a change in the TRV temperatures.

The results of step inputs to the slug velocity setpoint are shown in Fig. 20: As the slug velocity control system is realized as a cascaded loop, setpoint changes take several minutes to conclude while the “Delta-Controller” keeps the error between the two reactors below a deviation of 15% for 88.5% of the time. Between 200 and 400 s, the phase

separator pressure does not track its setpoint. This happens because the actuator, control valve VC12, is saturated (i.e., fully closed). Notably, this behavior is only present for a slug velocity setpoint of  $15 \text{ mm s}^{-1}$  and thus represents the lower boundary of the reactors’ operating range.

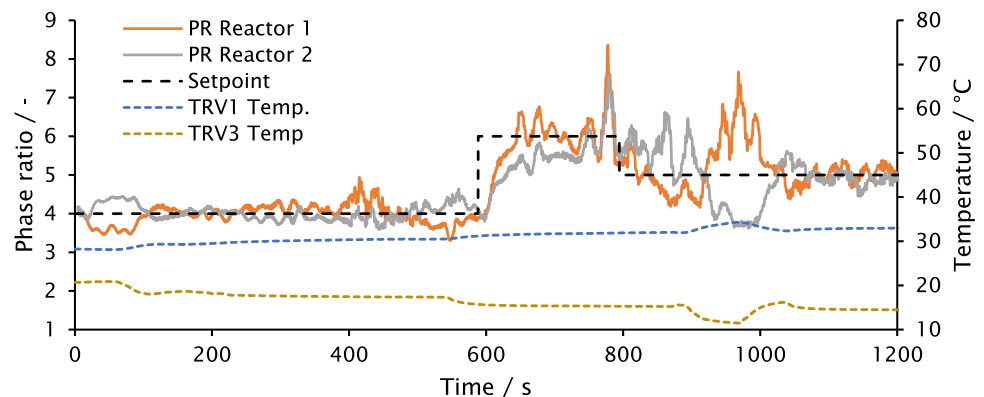
The inherent smoothing provided by the spatial processing appears to aid in providing suitable feedback signals to the controllers, however, in some cases a more rapid sensor response could bear the potential to improve both disturbance rejection as well as step responses.

### 3.2.3 Reactor startup performance

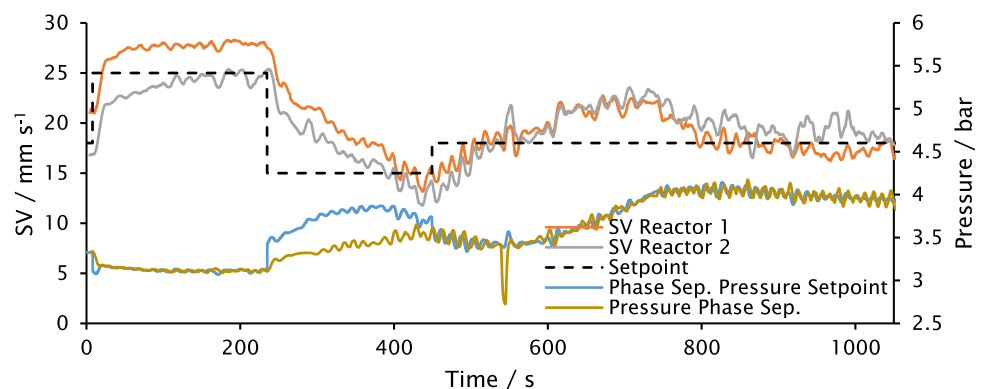
In addition to the control system’s ability to switch between and maintain setpoints, it is important the system can be brought into operation quickly and reliably to avoid unnecessary chemical waste.

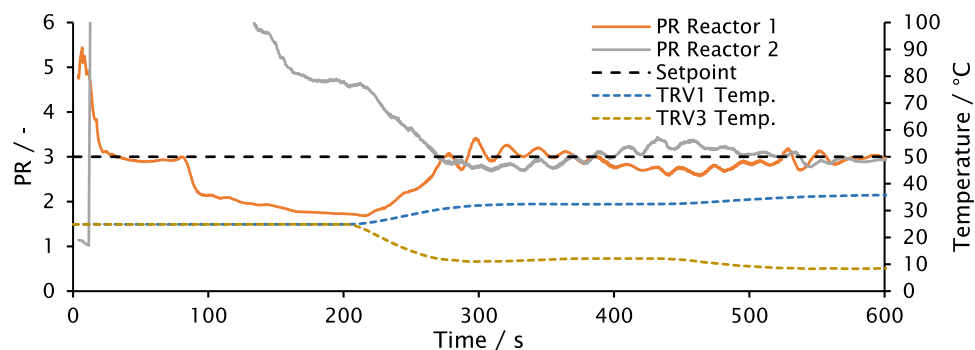
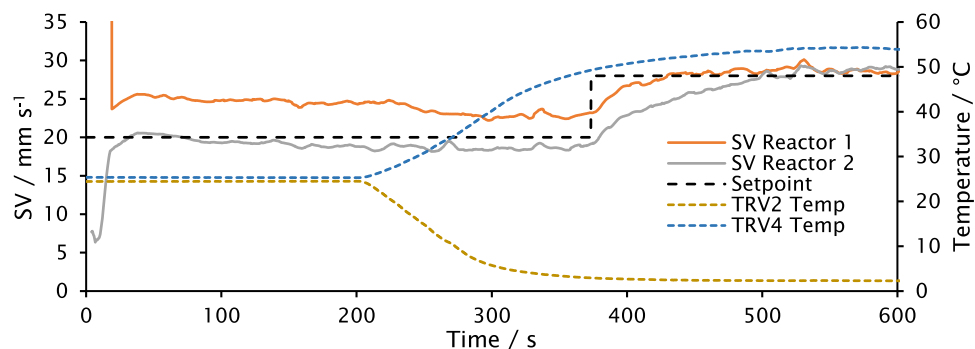
In general, the startup procedure consists of three steps: From the standby state (i.e., all shut-off valves are in the open position, but the flow is halted by the control valves), the controllers for the phase separator pressure, and level are switched on to start the flow of reactants. Once stable slug flow is present in both reactors, the slug velocity, phase ratio, and slug length control are switched on. From there, the controllers first attempt to bring the average phase ratio and slug length to their setpoints. Once the

**Fig. 19** Phase ratio (PR) setpoint changes



**Fig. 20** Slug velocity (SV) setpoint changes

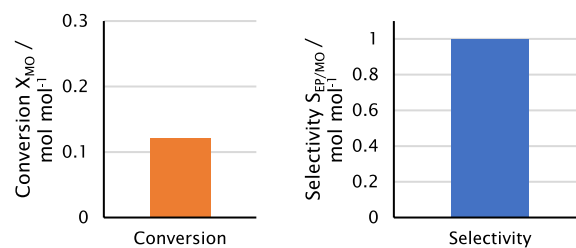


**Fig. 21** Phase ratio (PR) during startup procedure**Fig. 22** Slug velocity (SV) during startup procedure

setpoints of the averages are within their margins for 30 s, the “Delta-Controller” begins its operation (i.e., it enters its active state). Figures 21 and 22 show this process.

In general, the system behaves as intended and the startup process illustrates the actions of the “Delta-Controller” on step 3 well, where the individual reactor’s slug flow parameters slowly converge. However, Fig. 22 shows that at an average slug velocity of  $20 \text{ mm s}^{-1}$  the “Delta-Controller” is unable to fully converge the two reactors’ slug velocities, indicated by the TRV temperatures at the edges of their working range. The error approaches zero only once a higher slug velocity is set as reference. This is likely caused by the interference between the TRVs responsible for controlling phase ratio error and those controlling slug velocity error. If the aqueous phase is throttled, the phase ratio of the respective reactor must also increase, leading to “competition” or coupling between the control loops.

Throughout the step tests shown above, the computer vision slug flow sensor performs adequately. During the first stage of the startup procedure (halted flow) the sensor tends to return false measurements until the operating parameters are brought to within the sensor’s operating range, however by employing an open loop control scheme during this phase, the sensor’s limitations can be overcome.

**Fig. 23** Conversion and selectivity achieved in the SFCMR

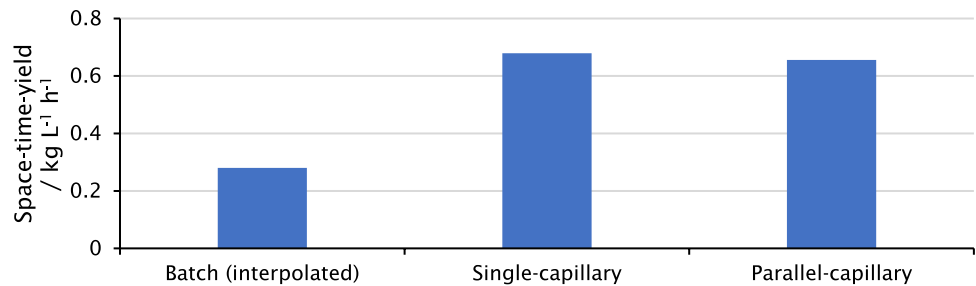
### 3.2.4 Reaction performance

To confirm the functionality of the SFCMR, an epoxidation reaction was carried out at a phase ratio of 3, a slug velocity of  $28 \text{ mm s}^{-1}$ , and a methyl oleate to catalyst ratio of  $80 \text{ g g}^{-1}$ . Because the parallel-capillary reactor includes the phase separator, which provides 4.40 ml of holdup volume to the organic phase, the samples are drawn after three residence times of the system to ensure the system operates at steady state. The results are displayed in Fig. 23.

Figure 24 shows the space–time-yields achieved in the parallel-capillary SFCMR, a single-capillary of the SFCMR as well as in stirred-batch mode.

Overall, the parallel-capillary SFCMR performs similarly to the single-capillary SFCMR, indicating that numbering-up was successful and reaction conditions in the two parallel-capillary reactors were like the reaction conditions in a single-capillary reactor.

**Fig. 24** Space–time-yields of the epoxidation of methyl oleate in batch, single-capillary slug flow, and parallel-capillary-slug flow modes.  $\Phi$ -3, MO:Cat (Methyloleat:Catalyst Ratio)—80 g g<sup>-1</sup>



## 4 Discussion

In the research conducted, a computer vision sensor was developed for the real-time extraction of slug length, slug velocity, and phase ratio from capillary liquid–liquid slug flows, utilizing video feeds. This development included the necessary post-processing algorithms. The sensor, with a demonstrated mean relative error of less than 3.5%, showcased its capability to monitor multiple capillaries simultaneously. It was then employed to control a parallelized and actively regulated dual-channel slug flow capillary microreactor setup. In a model reaction, the solvent-free epoxidation of methyl oleate was conducted with hydrogen peroxide and a phase-transfer catalyst, yielding a space–time yield of 0.679 kg L<sup>-1</sup> h<sup>-1</sup>.

When compared to previously reported results, the space–time yield obtained in this study was found to be less than the 1.29 kg L<sup>-1</sup> h<sup>-1</sup> yield reported by Gladius et al. [13], but notably higher than the 0.08 kg L<sup>-1</sup> h<sup>-1</sup> yield from traditional batch processes. These results indicated a careful balance between reactor performance and system maintenance as it was observed that the typical capillary clogging, frequently occurring with the catalyst system reported by Gladius et al., was significantly reduced in this study.

While the models for the spatial processing algorithm were trained on a single capillary, scaling up to multiple capillaries for testing and experimentation was successfully accomplished without the necessity for retraining. It is noteworthy to mention that the limiting factors for implementing the machine learning-based approaches are considered to be the capabilities of the GPU and the configuration of the camera.

In comparison to the common sensing techniques employed in liquid–liquid slug flow monitoring as described by Wolffenbuttel [23], the computer vision sensor developed in this study showcases a distinct potential for parallelization using centralized processing hardware. Wolffenbuttel pointed out that common techniques include impedance-, absorbance-, and reflectance sensors, and video imaging. The majority of these methods, including works of Xue et al. [24], Pietrasanta et al. [25] and Cierpka et al. [26] have inherent drawbacks such as

the necessity for complex image processing, the high cost of recording equipment, the use of electrodes that could potentially impact the flow, the addition of dyes that can alter fluid properties, or the use of modified optical fibre that can influence the capillary flow.

In light of other published works, such as the IR-based sensor developed by Vietinghoff et al. [22], it is of importance to recognize the balance between accuracy, latency, cost scaling, and the readiness for market deployment. Although the IR-based sensor can deliver slug flow measurements with relatively low latency, the use of decentralized microcontrollers provides little scope for sublinear cost scaling. On the other hand, video-based approaches, such as the one developed in this study, rely on a moving window, which introduces filter characteristics implying a loss of detail and increased latency. Nevertheless, the computer vision sensor developed in this study offers the potential for real-time monitoring of multiple channels, which may lead to a more efficient and economical solution for industrial-scale applications.

A potential drawback of this approach could be perceived as the requirement for a GPU to perform real-time inference on machine learning models. However, this study utilized a consumer-grade GPU (Nvidia RTX 2070), highlighting that the cost can be comparable or even lower than most of the sensing equipment currently available in the market for microfluidics. The increasing rate of advancements in computer vision suggest that the future could see the use of independent edge devices with computational performance sufficient for such tasks, further reducing the costs.

## 5 Conclusion and outlook

In conclusion, this study has successfully developed and validated a computer vision sensor for real-time monitoring of slug flow characteristics in capillary liquid–liquid systems. The sensor's ability to simultaneously monitor multiple capillaries with high accuracy provides valuable insights into the dynamics of slug flows and opens up new possibilities for process understanding and optimization in microreactor applications.

By integrating the computer vision sensor into a parallelized and actively regulated dual-channel slug flow capillary microreactor setup, the study achieved effective control and optimization of the solvent-free epoxidation reaction, resulting in a space–time yield of  $0.679 \text{ kg L}^{-1} \text{ h}^{-1}$ . This achievement demonstrates the practicality and reliability of the sensor in real-world applications, offering significant value in improving the efficiency and productivity of microreactor processes. The developed sensor offers several key advantages that contribute to its relevance and added value in the field. Firstly, its real-time monitoring capability provides instantaneous and continuous feedback on slug flow parameters, allowing for rapid adjustments and optimization of process conditions. This real-time feedback enables researchers and engineers to make informed decisions and take proactive measures to enhance the performance of microreactor systems.

Furthermore, the sensor's simultaneous monitoring of multiple capillaries enhances its versatility and scalability. This capability is particularly important in industrial-scale applications where multiple reactors operate in parallel. The ability to monitor and control multiple channels simultaneously streamlines the process, increases efficiency, and reduces the need for additional sensing equipment. The study also acknowledges certain limitations that warrant further investigation and improvement. One limitation is the challenge faced by the "Delta-Controller" in fully converging the slug velocities of the two reactors, especially at an average slug velocity of 20 mm/s. Interference between control loops is identified as a potential cause for this challenge. Addressing this limitation requires refining the control system, optimizing control strategies, and exploring alternative control schemes during start-up to ensure better convergence and control of slug flow parameters.

Additionally, the study recognizes the potential for false measurements during the initial halted flow phase. To overcome this limitation, the use of an open-loop control scheme during this phase is suggested. By employing open-loop control, the sensor's limitations during the start-up procedure can be mitigated, ensuring reliable and accurate slug flow measurements. Looking ahead, future research should focus on refining and advancing the computer vision sensor technology to unlock its full potential. Emphasis should be placed on improving the spatial processing algorithms to handle a wider range of data variations, including different camera heights and slug parameters. This will ensure the sensor's robustness and effectiveness in different experimental scenarios, enabling its broader application in diverse microreactor setups.

Moreover, exploring the integration of advanced hardware, such as independent edge devices with high computational performance, holds promise for enhancing the sensor's capabilities while reducing costs. By leveraging

consumer-grade GPUs and exploring alternative microchips like the Nvidia Jetson or Intel Neural Compute Stick, the development and implementation costs can be further reduced without compromising real-time inference capabilities. Furthermore, the video input from the developed sensors can be leveraged to train complex image/video recognition algorithms. This would enable the detection and control of intricate reaction parameters within the slug flow, including stagnant zones, vortices, irregular flow patterns, mixing rates, internal reactant/product circulation, mass transfer in/out of the slug, and capillary clogging. The integration of such advanced control mechanisms has the potential to significantly enhance the efficiency and performance of microreactor processes, opening up new opportunities for precise and adaptive reaction control.

Overall, the computer vision sensor developed in this study demonstrates significant potential for revolutionizing slug flow monitoring, control, and optimization in capillary liquid–liquid systems. By combining advancements in hardware, algorithmic improvements, and sophisticated control strategies, future research can further enhance the sensor's capabilities and drive innovation in the field of microreactor technology. Continued efforts in refining the sensor technology, addressing its limitations, and exploring new applications will contribute to the advancement of microreactor processes and their broader adoption in various industries.

**Acknowledgements** The authors acknowledge the contribution of Balazs Bordas, Rajasekhar Marella and Konstantin Jauk in helping with the with initial data collection and research on using Machine Learning and camera imaging to determine slug flow parameters.

**Author contributions** AWG: Conceptualization, methodology, investigation, writing—original draft preparation. JAM: Conceptualization, methodology, investigation, writing, and visualization. DWA: Supervision—review and editing.

**Funding** Open Access funding enabled and organized by Projekt DEAL. The authors acknowledge financial support by Deutsche Forschungsgemeinschaft and TU Dortmund University within the funding program Open Access Publishing.

**Data availability** The datasets generated during and/or analysed during the current study are available from the corresponding author on reasonable request.

## Declarations

**Conflict of interest** The authors declare no conflict of interest.

**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this



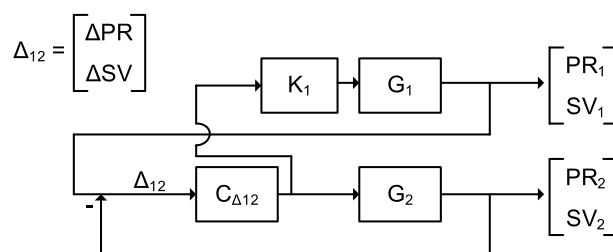
article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

## Appendix

### Appendix A: Controller design for interconnected capillary slug flow micro reactors

Because both reactors of the capillary SFCMR are connected at both the supply side as well as down-stream, they act as interconnected channels. As a practical means of addressing the coupling behaviour brought on by this, control of  $\varphi_i$  and  $v_{s,i}$  is decomposed into control of the average across all reactor channels, as well as control of the difference between both channels, using the “Delta-Controller” developed for this task, with a block diagram representation shown in Fig. 25. The average phase ratio  $\varphi$  between organic and aqueous phase across all channels is controlled by proportional-integral-controller (PI) QFIC2 using flow control valve VC8, which adjusts the relative flow resistance between organic and aqueous supply. The average slug velocity  $v_s$  is controlled in a cascaded control loop by PI-controller SIC2, which sets the pressure reference in settler-type phase separator B4. The pressure in phase separator B4, in turn, is controlled by PIC3 using valve VC12.

The temperatures for both pairs of TRVs, with the purpose of fine adjustment of the flow distribution, is set using the “Delta-Controller”. This controller arrangement effectively designates a main reactor and a follower. The “Delta-Controller” is made up of a PI controller  $C_{\Delta 12}$  with constant gain and an integrator, as well as a constant gain  $K_1$  which is used to extend the controller's working range. A more detailed analysis of coupling effects within multichannel liquid–liquid slug flow setups, the working principles of the slug generator, as well as the TRVs used here is provided by Arsenjuk et al. and shall not be discussed further [12, 21]



**Fig. 25** Block diagram representation of the “Delta-Controller”, where  $C_{\Delta 12}$  is of the form of a standard PI-controller with a proportional gain and an integrator,  $K_1$  is a simple gain,  $G_1$  and  $G_2$  represent the plant while  $\Delta_{12}$  is the difference in  $\varphi$  and  $v_s$  between the two channels, respectively

### Appendix B: Preparation of the epoxidation catalyst

In a procedure adapted from Maiti et al. [32] to produce the *Venturello Complex*  $Q_3[PW_{12}O_{40}(O)]$ , 2.88 g of TPA (Carl Roth GmbH and Co. KG) are added to 12 ml of 50%-w  $H_2O_2$  (Carl Roth GmbH and Co. KG) and stirred until dissolved. The resulting solution is diluted with 18 ml of distilled water and stirred for an additional ten minutes. To the solution, 2.42 g of Q (Alfa Aesar) in 60 ml ethyl acetate are added dropwise over five minutes and stirred for 30 min at room temperature and a stirring speed of 800 rpm. After settling and separation of the phases, the organic phase is desiccated over anhydrous magnesium sulfate for 30 min. In a vacuum distillation apparatus, the ethyl acetate is evaporated from the organic phase at 40°C and collected for reuse. The active catalyst species appears as a highly viscous liquid with a light-yellow hue and is stored at 6°C to prevent deterioration. As the catalyst is prepared with a two-to-one excess of Q, free Q is present in the product.

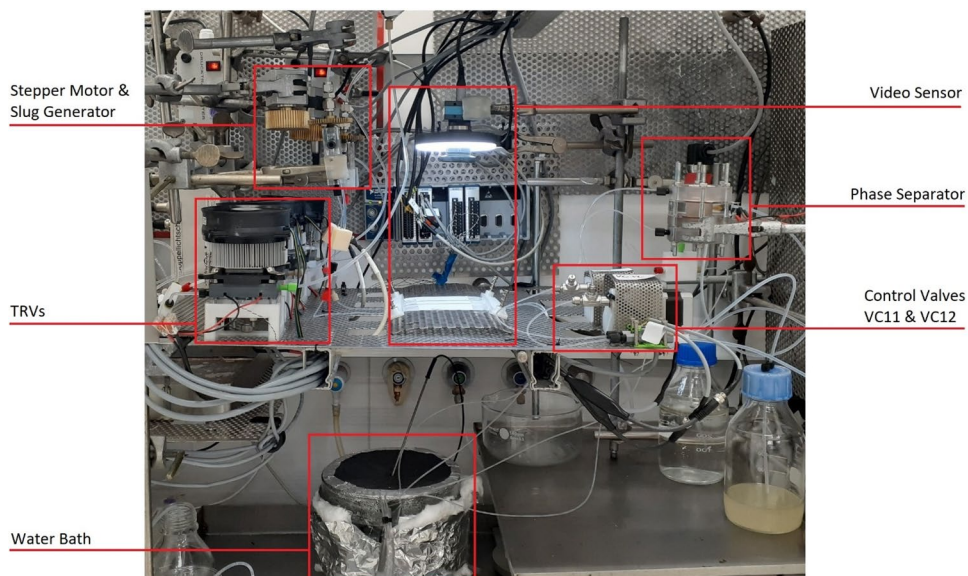
The catalyst is analysed by ATR-FTIR. Peaks at 3000–2800  $cm^{-1}$  and 1465  $cm^{-1}$  are associated with the methyl groups of Q, while peaks at 1077  $cm^{-1}$  (P-O) at 974  $cm^{-1}$  (W=O) and 816  $cm^{-1}$  (W–O–W) show that all components of the catalyst are present [33, 42, 43]. In addition, trial reactions are conducted in stirred batches: 3 g of technical grade methyl oleate, 1.27 g of 35%-w  $H_2O_2$  and 0.0676 g of catalyst are heated to 60 °C and stirred at 400 rpm for one hour. The reaction is quenched using distilled ice water, which is added to the mixture rapidly. The products are analysed using an Agilent 7693A GC-FID device equipped with an Agilent 19091J-413 column. The methyl palmitate contained in the technical grade methyl oleate reactant is used as a reference to determine reaction progress. Under these conditions a conversion of methyl oleate of 84.4% at a selectivity toward 9,10-Epoxystearic acid methyl ester

of 89.9% is observed, falling in line with the results obtained in other works [15]. As a control, by adding no catalyst, the same procedure shows no measurable conversion, further confirming the produced catalyst's efficacy.

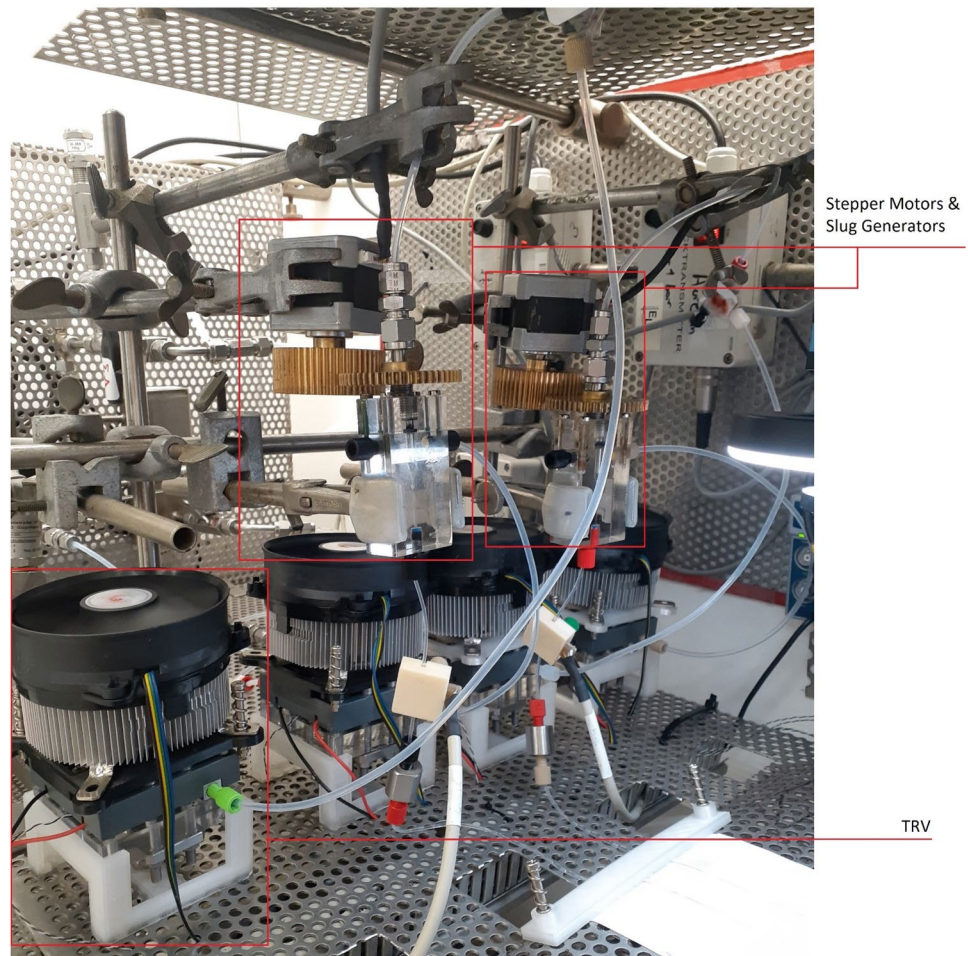
### Appendix C: Additional photo documentation

Shown below in Figs. 26 and 27 are the Overview of the experimental setup and the zoomed in View of the TRVs and the Stepper Motor and Slug Generator Setup respectively.

**Fig. 26** Overview of experimental setup



**Fig. 27** Zoomed in View of the TRVs and the Stepper Motor and Slug Generator Setup



## Appendix D: Robustness and reliability of the proposed sensor

As explained in the main text, many different models were tried upon to test the reliability, durability and scalability of the sensing technology being developed. Various models were tried to test out various environmental/physical problems that the system might face in real world application. Some of the experiments conducted and the results obtained have been attached here. It is also then explained why (Yolov4-416) model was chosen as the main challenger.

Figure 28 shows the number of capillaries that could be measured at different heights using different models. It also shows the maximum number of capillaries that could be detected and measured, through extrapolation and actual testing on the stand. Figure 28 also shows that when fed with video streams larger than intended, these models do not perform well over greater heights compared to models fed with their intended sizes. This is expected, but it also shows that the models can be flexible

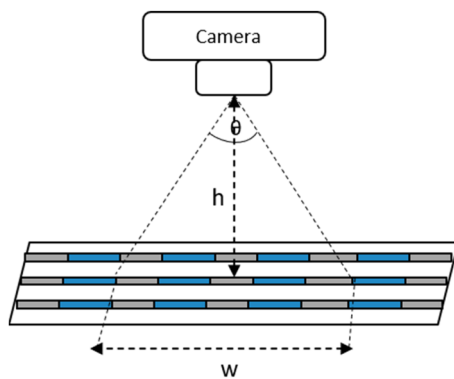
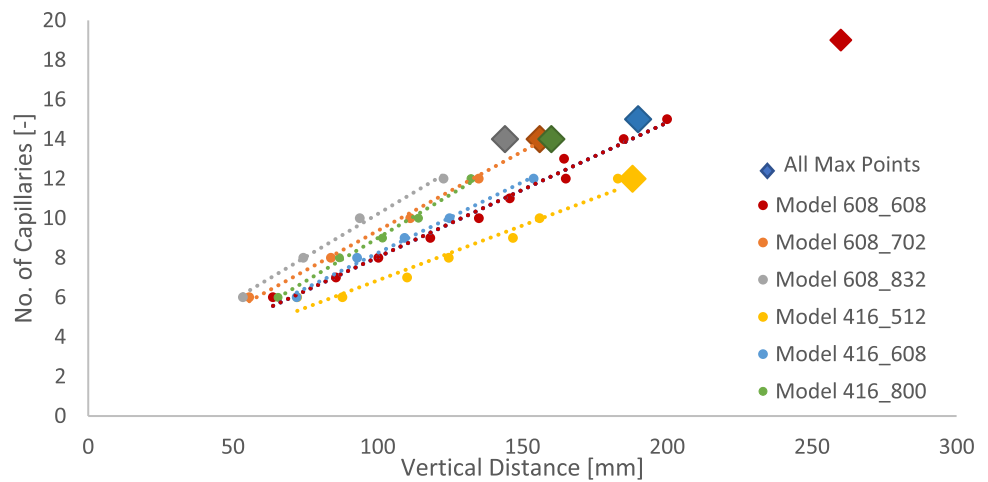
with decent performance without any additional training. Points marked with '◆' are the distances where the model fails to detect even a single slug or has too many false detections. The model fed with the size of the images from the training data set performs much better over its counterparts.

Also seen in Fig. 28 is that with all object detector models, there exists a linear relationship between height and the number of capillaries. This can be explained by the nature of the angle formed by the field of view of the camera, as shown in Fig. 29. The experiments are done at angles where Eq. 10 is approximately linear, i.e., the value of  $\tan\theta$  changes linearly and not exponentially, as expected.

$$h = w \times \tan\theta \quad (10)$$

As expected, regardless of the type of model the larger the dimension of the video stream, the higher the number of capillaries it could 'fit' into its frame. However, to truly see the non-linear relationship between vertical distance

**Fig. 28** Vertical distance of the camera from capillaries vs. No. of capillaries for various ML models

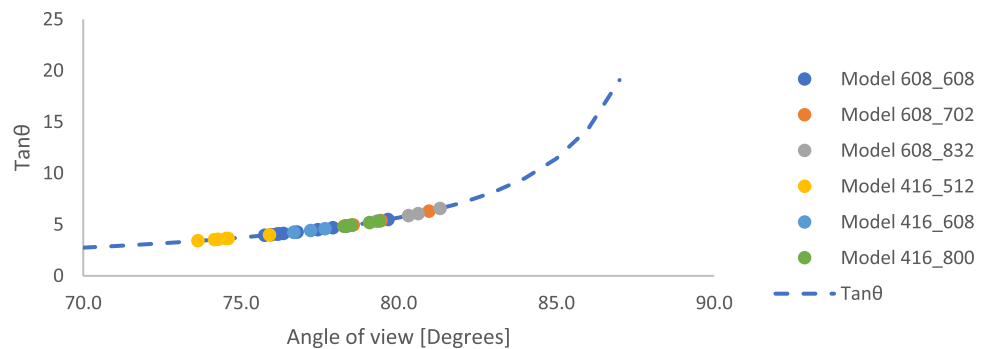


**Fig. 29** Angle of view from the perspective of the camera's objective lens

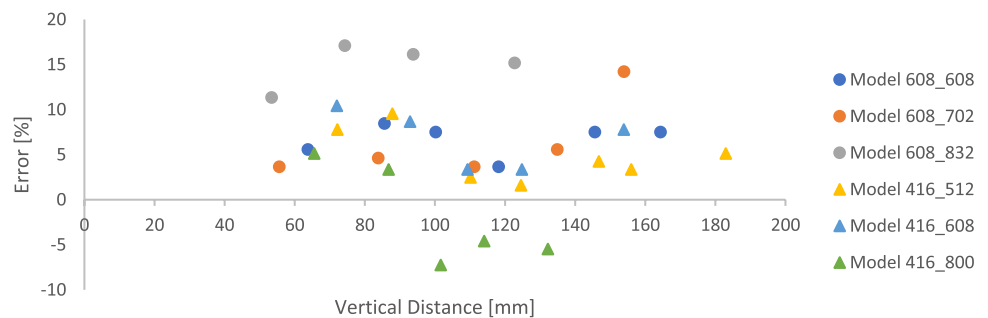
and number of capillaries, either a very wide field-of-view of the camera must be used i.e., a model trained on higher image size or a model trained with a lower image size must be fed higher image size data, with a trade-off of detection accuracy.

Figure 30 explains the linear relationship between the vertical distance of the camera and the number of capillaries for different models at different heights as well as the 'horizon' of linearity, where the angle of view shows strong non-linearity only beyond 82°. For example, Model 608\_702, which uses Yolov4-608, as the base model, where the input image of size 702 × 702 is resized to 608 × 608, run at different heights correspond to different angles of views, hence different values of Tanθ. Strong non-linearity in the increase in the number of capillaries will occur if the

**Fig. 30** Tanθ vs. Angle of views for different camera positions and models



**Fig. 31** Error deviation of slug length prediction for different models





angle exceeds  $82^\circ$ , after which even if a greater number of capillaries are brought into the field of view, the accuracy of detection is still unknown. If a higher field of view is necessary, new data will have to be recorded at higher pixel densities, e.g.,  $1024 \times 1024$  at the same height, and new models will have to be trained on this data. Theoretically, the limiting factor in this experiment is the camera's objective lenses' field of view.

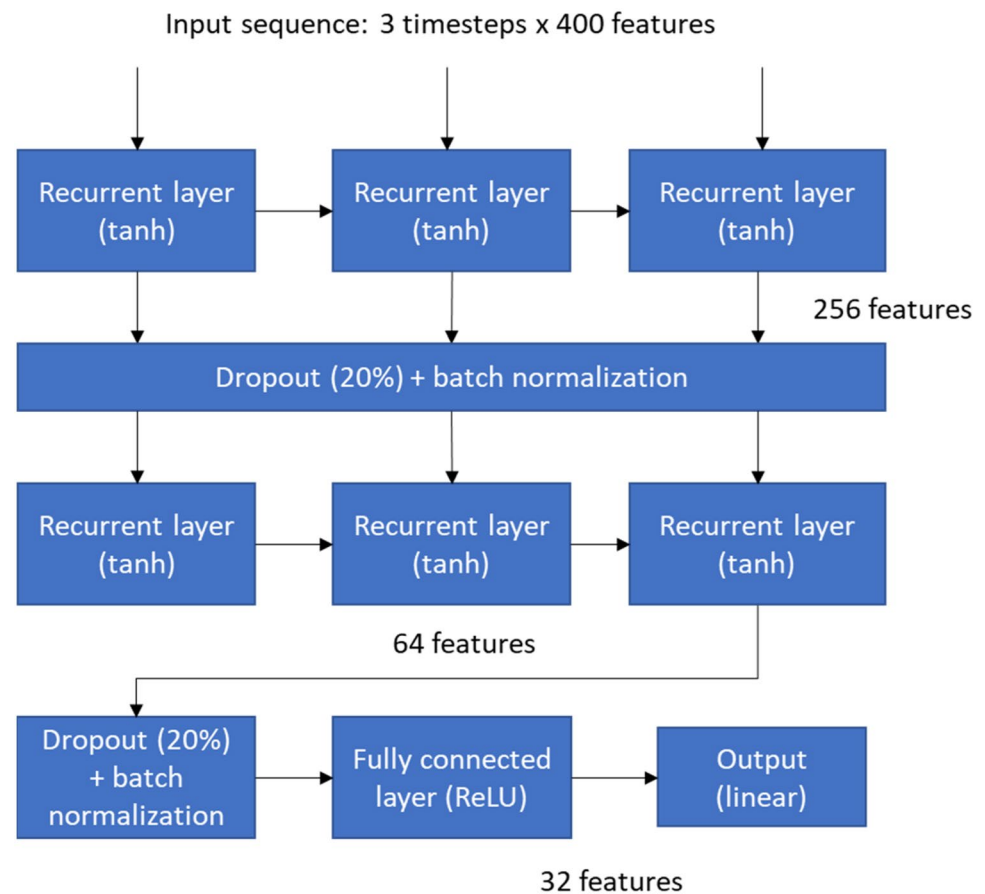
Complementing the results from Figs. 28, 29, 30, 31 shows how accurate the object detector is at different heights for each model (and its combinations). Most of the models predict the slugs within an error margin of  $\pm 5\%$ . A general trend that can be observed is that error is lower at vertical distances of approximately 120 mm. This could be attributed to the fact that the original training data is recorded at approximately this elevation. An unexpected trend is that even the lower dimension model (Yolov4-416) showed good accuracy.

## Appendix E: Training and evaluation of LSTM slug velocity detector model candidates

The first model candidate is illustrated in Fig. 32 for the 3-timestep case. To facilitate the discussion about different model candidates, a simple nomenclature is introduced to describe their structure. The model illustrated in Fig. 32 can be written as 3-RNN (128-64)-FC(32), where the first number refers to the number of timesteps and the numbers in brackets denote the output dimensions of the corresponding layers. Dropout is only used after the recurrent layers while batch normalization takes place after all but the last fully connected layer.

During training, the root means the squared error is applied as the loss function to penalize outlying predictions but the mean absolute error is used as the final performance metric. The training procedure is terminated if the validation loss did not decrease for ten consecutive training epochs. As before, the Adam optimizer is used with  $\eta = 0.001$  and a batch size of 32.

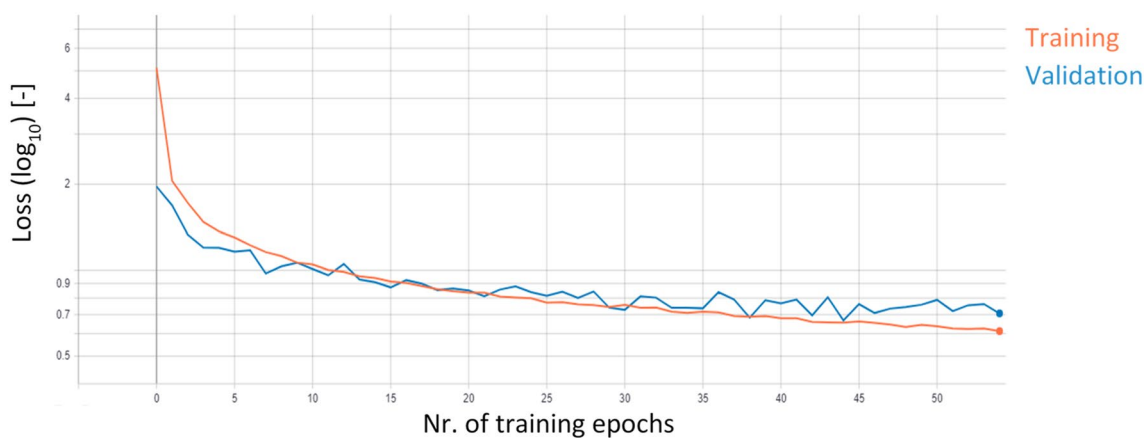
**Fig. 32** Structure of the first recurrent network candidate with three timesteps





**Table 2** Trained RNN model candidates along with their size and performance

Structure	Nr. of trainable weights [-]	Mean abs. error on validation set [mm s <sup>-1</sup> ]
3-RNN(256-64)-FC(32)	191489	0.997
5-RNN(256-64)-FC(32)	191489	0.808
5-LSTM(256-64)-FC(32)	757697	0.564
5-LSTM(256-128)-FC(64-32-16)	881729	0.491
5-LSTM(128)-FC(64-32-16)	282177	0.564
5-LSTM(128-64)-FC(32)	322753	0.641
5-LSTM(256)-FC(32)	681537	0.696
5-LSTM(128)-FC(128-128-64-32-16)	315713	0.618
5-LSTM(256-128-64)-FC(32)	922369	0.64
5-LSTM(256-128)-FC(64-32)	881153	0.539



**Fig. 33** Evolution of the training and validation losses of 5-LSTM(256-128)-FC(64-32-16)

Table 2 summarizes the structure of the model candidates along with their size in terms of trainable weights and performance on the validation set. The funnel-like network arrangements in terms of layer output dimensions are mainly inspired by the YOLO network. Following this pattern proved to be useful in providing some structure to the otherwise arbitrary search after suitable layer sizes and permutations.

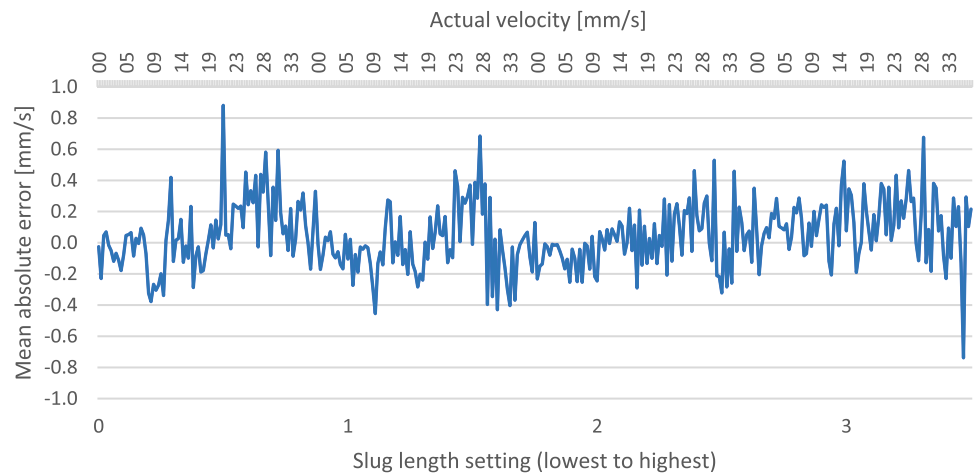
The first two basic recurrent networks only differ in the number of timesteps in the input sequence. While the improvement is not dramatic, a decrease in the mean absolute error of roughly 0.2 mm/s suggests that the assumption about skipped video frames having less influence when more frames are used for detection, may hold, therefore the use of 5 timesteps is adopted in further model candidates.

The existence of missing frames encouraged the trying of LSTMs instead of basic recurrent layers, as it is expected that the gating mechanisms could help the velocity detector learn to ignore such irregularities. The third model

candidate with LSTM layers but with the same layer output dimensions and depth further reduced the mean absolute error by 0.24 mm/s. 5-LSTM(256-128)-FC(64-32-16), highlighted with red in Table 2, is the only network that achieves a mean absolute error less than 0.5 mm/s on the validation set. The performance of 5-LSTM(128)-FC(64-32-16) is also impressive when the much more compact network size is considered. Five further model candidates with inferior performances are listed in the second half of the table. Figure 33 shows the evolution of the root mean squared error values on the training and validation set during the training of 5-LSTM(256-128)-FC(64-32-16), which is selected as the final velocity detector.

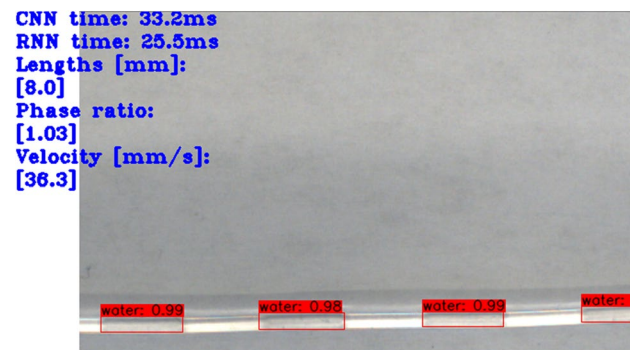
Figure 34 shows the distribution of mean absolute errors over the 352 recordings concerning the actual velocity (upper horizontal axis) and grouped by slug length settings (lower horizontal axis), revealing no patterns that would indicate the presence of systematic error.

**Fig. 34** Error distribution of 5-LSTM(256-128)-FC(64-32-16) on the complete dataset for the actual (calculated) velocity values and grouped by slug length settings

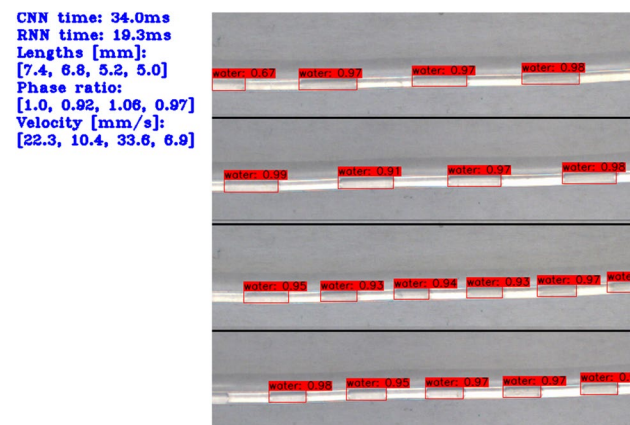


### Appendix F: Investigation of limitations in object detection networks for capturing capillary slug flow characteristics

This appendix presents the experimental evaluation and analysis of object detection networks for capturing



**Fig. 35** Detection on a single channel



**Fig. 36** Detection on four channels

CNN time: 33.8ms  
RNN time: 30.0ms  
Lengths [mm]:  
[7.2, 6.7, 5.2, 4.9, 5.2, 4.4, 8.7, 4.4]  
Phase ratio:  
[1.02, 0.93, 1.17, 0.98, 1.07, 1.17, 1.03, 1.01]  
Velocity [mm/s]:  
[22.2, 10.8, 33.6, 7.0, 17.6, 1.7, 27.5, 5.7]



**Fig. 37** Detection on eight channels

the unique characteristics of capillary slug flows. The limitations and challenges faced by these networks in accurately differentiating between closely spaced slugs, identifying irregularly shaped slugs, and estimating slug lengths and velocities in complex flow patterns, as highlighted by the reviewer, are explored. A series of experiments was conducted across multiple capillaries with different compositions of immiscible substances to address these aspects. The obtained results shed light on the robustness and reliability of the proposed method. Included are figures that visually represent the experimental findings, providing a comprehensive evaluation of the performance of object detection networks in the context of capillary slug flow detection.

Figures 35, 36 and 37 show the sensor working on 1, 4, and 8 channels, respectively. The indicators “CNN time” and “RNN time” show the overall execution time for the object and velocity detector networks in the test environment. Owing to the way the YOLO class of models infer from images; it is expected that increasing the number of objects in an image does not increase the inference time. This can be seen in Figs. 36 and 37

as execution time barely increases with the number of channels, which can be attributed to the batching of input sequences of the RNN and the high parallel computing capability of the used GPU architecture optimised for CNNs and Computer Vision tasks in general.

## References

1. Elzinga D, Baritaud M, Bennett S, Burnard K, Pales, Philibert C, Cuenot F, D'Ambrosio D, Dulac J, Heinen S et al. (2014) Energy technology perspectives 2014: harnessing electricity's potential. <https://iea.blob.core.windows.net/assets/f97efce0-cb12-4caa-ab19-2328eb37a185/EnergyTechnologyPerspectives2014.pdf>. Accessed 06 Jul 2023
2. Kim Y-H, Park LK, Yiacoymi S, Tsouris C (2017) Modular chemical process intensification: a review. *Annu Rev Chem Biomol Eng* 8:359–380. <https://doi.org/10.1146/annurev-chembioeng-060816-101354>
3. Tam C, Baron R, Gielen D, Taylor M, Taylor P, Trudeau N, Patel M, Saygin D (2009) Energy technology transitions for industry: strategies for the next industrial revolution, OECD/IEA
4. Cortes Garcia GE, van der Schaaf J, Kiss AA (2017) A review on process intensification in HiGee distillation. *J Chem Technol Biotechnol* 92(6):1136–1156. <https://doi.org/10.1002/jctb.5206>
5. Stankiewicz A, Moulijn JA (2003) Re-engineering the chemical processing plant: process intensification. CRC Press, London
6. Aubin J, Ferrando M, Jiricny V (2010) Current methods for characterising mixing and flow in microchannels. *Chem Eng Sci* 65(6):2065–2093. <https://doi.org/10.1016/j.ces.2009.12.001>
7. van Gerven T, Stankiewicz A (2009) Structure, energy, synergy, time. The fundamentals of process intensification. *Ind Eng Chem Res* 48(5):2465–2474. <https://doi.org/10.1021/ie801501y>
8. Lerou JJ, Tonkovich AL, Silva L, Perry S, McDaniel J (2010) Micro-channel reactor architecture enables greener processes. *Chem Eng Sci* 65(1):380–385. <https://doi.org/10.1016/j.ces.2009.07.020>
9. Kashid MN, Gerlach I, Goetz S, Franzke J, Acker JF, Platte F, Agar DW, Turek S (2005) Internal circulation within the liquid slugs of a liquid–liquid slug-flow capillary microreactor. *Ind Eng Chem Res* 44(14):5003–5010. <https://doi.org/10.1021/ie0490536>
10. Kashid MN, Renken A, Kiwi-Minsker L (2011) Gas–liquid and liquid–liquid mass transfer in microstructured reactors. *Chem Eng Sci* 66(17):3876–3897. <https://doi.org/10.1016/j.ces.2011.05.015>
11. Taha T, Cui ZF (2004) Hydrodynamics of slug flow inside capillaries. *Chem Eng Sci* 59:1181–1190. <https://doi.org/10.1016/j.ces.2003.10.025>
12. Arsenjuk L, Asshoff M, Kleinheider J, Agar DW (2020) A device for continuous and flexible adjustment of liquid–liquid slug size in micro-channels. *J Flow Chem* 10(2):409–422. <https://doi.org/10.1007/s41981-019-00064-7>
13. Gladius AW, Vondran J, Ramesh Y, Seidensticker T, Agar DW (2021) Slug flow as tool for selectivity control in the homogeneously catalysed solvent-free epoxidation of methyl oleate. *J Flow Chem*. <https://doi.org/10.1007/s41981-021-00199-6>
14. Anastas PT, Warner JC (2000) Green chemistry. Oxford University Press, Oxford, p 135
15. Kozhevnikov IV, Mulder GP, Steverink-de Zoete MC, Oostwal MG (1998) Epoxidation of oleic acid catalyzed by peroxo phosphotungstate in a two-phase system. *J Mol Catal A Chem* 134(1–3):223–228. [https://doi.org/10.1016/S1381-1169\(98\)00039-9](https://doi.org/10.1016/S1381-1169(98)00039-9)
16. Kashid MN, Renken A, Kiwi-Minsker L (2015) Microstructured devices for chemical processing. Wiley, Weinheim
17. Kashid MN, Gupta A, Renken A, Kiwi-Minsker L (2010) Numbering-up and mass transfer studies of liquid–liquid two-phase microstructured reactors. *Chem Eng J* 158(2):233–240. <https://doi.org/10.1016/j.cej.2010.01.020>
18. Kockmann N, Gottsponer M, Roberge DM (2011) Scale-up concept of single-channel microreactors from process development to industrial production. *Chem Eng J* 167(2–3):718–726. <https://doi.org/10.1016/j.cej.2010.08.089>
19. Kockmann N, Roberge DM (2011) Scale-up concept for modular microstructured reactors based on mixing, heat transfer, and reactor safety. *Chem Eng Process* 50(10):1017–1026. <https://doi.org/10.1016/j.cep.2011.05.021>
20. Zhang J, Wang K, Teixeira AR, Jensen KF, Luo G (2017) Design and scaling up of microchemical systems: a review. *Annu Rev Chem Biomol Eng* 8:285–305. <https://doi.org/10.1146/annurev-chembioeng-060816-101443>
21. Arsenjuk L, Vietinghoff N, Gladius AW, Agar DW (2020) Actively homogenizing fluid distribution and slug length of liquid–liquid segmented flow in parallelized microchannels. *Chem Eng Processing Process Intensif* 156:108061. <https://doi.org/10.1016/j.cep.2020.108061>
22. Vietinghoff N, Lungrin W, Schulzke R, Tilly J, Agar DW (2020) Photoelectric sensor for fast and low-priced determination of bi- and triphasic segmented slug flow parameters. *Sensors (Basel, Switzerland)* 20(23):6948. <https://doi.org/10.3390/s20236948>
23. Wolffenbuttel BMA et al (2002) Novel method for non-intrusive measurement of velocity and slug length in two- and three-phase slug flow in capillaries. *Meas Sci Technol* 13:1540–1544. <https://doi.org/10.1088/0957-0233/13/10/305>
24. Xue T, Wang Q, Li C (2019) Analysis of horizontal slug translational velocity based on the image processing technique. pp. 1–5
25. Pietrasanta L, Mameli M, Mangini D, Georgoulas A, Michè N, Filippeschi S, Marengo M (2020) Developing flow pattern maps for accelerated two-phase capillary flows. *Exp Thermal Fluid Sci* 112:109981. <https://doi.org/10.1016/j.expthermflusci.2019.109981>
26. Cierpka C, Kähler CJ (2012) Particle imaging techniques for volumetric three-component (3D3C) velocity measurements in microfluidics. *J Visualiz* 15(1):1–31. <https://doi.org/10.1007/s12650-011-0107-9>
27. Redmon J, Farhadi A (2018) YOLOv3: an incremental improvement. *Comput Vision Pattern Recognit*. <https://doi.org/10.48550/arXiv.1804.02767>
28. Bochkovskiy A, Wang C-Y, Liao H-YM (2020) YOLOv4: optimal speed and accuracy of object detection. *Arxiv*. <https://doi.org/10.48550/arXiv.2004.10934>
29. Girshick R, Donahue J, Darrell T, Malik J (2014) Rich feature hierarchies for accurate object detection and semantic segmentation, *IEEE*, 580–587, <https://doi.org/10.48550/arXiv.1311.2524>.
30. Haykin S (2009) Neural networks and learning machines, 3rd ed., Pearson International Edition
31. Redmon J, Divvala S, Girshick R, Farhadi A (2015) You only look once: unified real-time object detection. *Comput Vision Pattern Recognit*. <https://doi.org/10.48550/arXiv.1506.02640>
32. Maiti SK, Snaveley WK, Venkatasubramanian P, Hagberg EC, Busch DH, Subramaniam B (2019) Reaction engineering studies of the epoxidation of fatty acid methyl esters with venturello complex. *Ind Eng Chem Res* 58(7):2514–2523. <https://doi.org/10.1021/acs.iecr.8b05977>
33. Wang M-L, Huang T-H (2004) Kinetic study of the epoxidation of 1,7-octadiene under phase-transfer-catalysis conditions. *Ind Eng Chem Res* 43(3):675–681. <https://doi.org/10.1021/ie030253b>
34. Raina R, Madhavan A, Ng AY (2009) Large-scale deep unsupervised learning using graphics processors, in *ICML '09: Proceedings of the 26th Annual International Conference on Machine*

- Learning, New York, USA, ACM Press, 2009, pp. 1–8, <https://doi.org/10.1145/1553374.1553486>.
35. Gu J, Wang Z, Kuen J, Ma L, Shahroudy A, Shuai B, Liu T, Wang X, Wang L, Wang G, Cai J, Chen T (2017) Recent advances in convolutional neural networks. Arxiv. <https://doi.org/10.48550/arXiv.1512.07108>
  36. Mahony NO, Campbell S, Carvalho A, Harapanahalli S, Velasco-Hernandez G, Krpalkova L, Riordan D, Walsh J (2019) Deep learning vs. traditional computer vision. Comput Vision Pattern Recognit. <https://doi.org/10.48550/arXiv.1910.13796>
  37. Redmon J, Farhadi A (2016) YOLO9000: Better, Faster, Stronger. Comput Vision Pattern Recognit. <https://doi.org/10.48550/arXiv.1612.08242>
  38. Jiang Z, Zhao L, Li S, Jia Y (2020) Real-time object detection method based on improved YOLOv4-tiny. Arxiv. <https://doi.org/10.48550/arXiv.2011.04244>
  39. Hochreiter S, Schmidhuber J (1997) Long short-term memory. Neural Comput 9(8):1735–1780. <https://doi.org/10.1162/neco.1997.9.8.1735>
  40. Gers FA, Schmidhuber J, Cummins F (1999) Learning to forget: continual prediction with LSTM. Neural Computat. <https://doi.org/10.1162/089976600300015015>
  41. Greff K, Srivastava RK, Koutnik J, Steunebrink BR, Schmidhuber J (2016) LSTM: a search space odyssey. Trans Neural Networks Learning Syst 28(10):2222–2232. <https://doi.org/10.1109/TNNLS.2016.2582924>
  42. Vahur S, Teearu A, Peets P, Joosu L, Leito I (2016) ATR-FT-IR spectral collection of conservation materials in the extended region of 4000–80 cm<sup>-1</sup>. Anal Bioanal Chem 408(13):3373–3379. <https://doi.org/10.1007/s00216-016-9411-5>
  43. Zhang H, Zhu B, Xu Y (2006) Composite membranes of sulfonated poly(phthalazinone ether ketone) doped with 12-phosphotungstic acid (H3PW12O40) for proton exchange membranes. Solid State Ionics 177(13–14):1123–1128. <https://doi.org/10.1016/j.ssi.2006.05.010>

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.