Max Planck Graduate Center
mit der Johannes Gutenberg-Universität

JOHANNES GUTENBERG UNIVERSITÄT MAINZ

“Spatio-temporal deep learning for modeling dynamic drop-surface interactions”

Dissertation

for the award of the academic degree of

“Doctor rerum naturalium” (Dr. rer. nat.) of the faculties:

08 - Physics, Mathematics and Computer Science

09 - Chemistry, Pharmacy, Geography and Geosciences

10 - Biology

University Medicine

submitted by

Sajjad Shumaly

Mainz, February 2026

Supervisor:
Prof. Dr. Hans-Jürgen Butt

Second supervisor:
Prof. Dr. Michael Wand

Day of the oral examination:
07.05.2026

Abstract

Sliding water drops are a familiar everyday phenomenon, for example on windows, but they also play an important role in many industrial processes. They serve as sensitive probes of wetting, adhesion, friction, and electrostatic charges, yet quantitative analysis remains difficult. In particular, friction forces depend on the contact-line width and dynamic advancing and receding contact angles. Measuring these quantities across the sliding path is difficult because front-view, high-resolution imaging requires complex optics and restricts the observable area.

This dissertation presents a single-view quantitative drop measurement framework that extracts drop geometry and dynamics from high-speed side-view videos using a combination of signal processing, computer vision, machine learning, and time-series modeling. It enables automated analysis without additional cameras or mirror-based front-view setups and makes it possible to track drop metrics across the full sliding path.

First, the 4-segment super-resolution optimized-fitting (4S-SROF) method couples an Efficient Sub-Pixel Convolutional Network with an optimized polynomial fitting strategy to reconstruct high-resolution drop contours and extract dynamic contact angles from low-resolution videos. The method improves contact-angle accuracy by about 20% for angles below 90° and 30% above 90°, while remaining computationally efficient for large datasets.

Second, the dissertation formulates front-view contact-line width estimation as a temporal inference problem from side-view measurements. Using water and water–glycerol drops on surfaces with controlled chemical and topographic patterns, an LSTM model achieves an RMSE of about 67 μm (approximately 2.4% relative error) and reconstructs drop width continuously along the sliding path, avoiding the mirror or second-camera limitation of front-view imaging.

Third, hand-crafted features are replaced by end-to-end spatiotemporal representation learning using a CNN-Transformer architecture that operates on short video sequences and velocity. A position-invariant video processing pipeline keeps the drop centered in a sliding spatial window and reduces memory and computation by over 80%. A custom BlurVGG8-ConvTran model with low-dimensional absolute positional encoding achieves about 48 μm error (approximately 1.7% relative), remains robust under surface defects and imaging perturbations, and provides Grad-CAM visualizations to interpret salient regions.

Overall, the dissertation delivers a unified, data-driven pipeline for measurement from single-view scientific videos. The approach simplifies experimental instrumentation and enables precise, automated estimation of drop dynamics. It further allows researchers to monitor drop width continuously along the full sliding path, overcoming the field-of-view limitations of conventional front-view imaging. Beyond wetting, the proposed methods provide broadly applicable tools for learning-based measurement under practical imaging constraints.

Zusammenfassung

Gleitende Wassertropfen sind ein vertrautes Alltagsphänomen, zum Beispiel an Fensterscheiben, spielen aber auch in vielen industriellen Prozessen eine wichtige Rolle. Sie dienen als empfindliche Sonden für Benetzung, Adhäsion, Reibung und elektrostatische Aufladung, dennoch bleibt ihre quantitative Analyse schwierig. Insbesondere hängen Reibungskräfte von der Kontaktlinienbreite sowie von dynamischen vorlaufenden und rücklaufenden Kontaktwinkeln ab. Die Messung dieser Größen entlang der Gleitstrecke ist schwierig, da eine hochauflösende Bildgebung in Frontansicht eine komplexe Optik erfordert und den beobachtbaren Bereich einschränkt.

Diese Dissertation stellt ein Einzelansichts-Framework zur quantitativen Tropfenmessung vor, das Tropfengeometrie und -dynamik aus Hochgeschwindigkeits-Seitenansichtsvideos extrahiert, unter Verwendung einer Kombination aus klassischer Bild- und Signalverarbeitung, Computer Vision, Maschinellem Lernen und Zeitreihenmodellierung. Es ermöglicht eine automatisierte Analyse ohne zusätzliche Kameras oder spiegelbasierte Frontansichtsaufbauten und macht es möglich, Tropfenmetriken entlang der gesamten Gleitstrecke zu verfolgen.

Zunächst koppelt die 4-segment super-resolution optimized-fitting (4S-SROF)-Methode ein Efficient Sub-Pixel Convolutional Network mit einer optimierten Polynom-Anpassungsstrategie, um hochauflösende Tropfenkonturen zu rekonstruieren und dynamische Kontaktwinkel aus niedrig aufgelösten Videos zu extrahieren. Die Methode verbessert die Genauigkeit der Kontaktwinkelbestimmung um etwa 20% für Winkel unter 90° und um 30% für Winkel über 90°, bleibt dabei jedoch recheneffizient für große Datensätze.

Zweitens formuliert die Dissertation die Schätzung der Kontaktlinienbreite in Frontansicht als ein zeitliches Inferenzproblem auf Basis von Messungen aus der Seitenansicht. Unter Verwendung von Wasser- und Wasser–Glycerol-Tropfen auf Oberflächen mit kontrollierten chemischen und topographischen Mustern erreicht ein LSTM-Modell einen RMSE von etwa 67 μm (ungefähr 2,4% relativer Fehler) und rekonstruiert die Tropfenbreite kontinuierlich entlang der Gleitstrecke, wodurch die Einschränkung der Frontansichtsbildgebung durch Spiegel oder eine zweite Kamera vermieden wird.

Drittens werden handgefertigte Merkmale durch End-to-End spatiotemporales Repräsentationslernen mittels einer CNN-Transformer-Architektur ersetzt, die auf kurzen Videosequenzen und der Geschwindigkeit arbeitet. Eine positionsinvariante Videoverarbeitungspipeline hält den Tropfen in einem gleitenden räumlichen Fenster zentriert und reduziert Speicherbedarf und Rechenaufwand um über 80%. Ein maßgeschneidertes BlurVGG8-ConvTran-Modell mit niedrigdimensionaler absoluter Positionskodierung erreicht einen Fehler von etwa 48 μm (ungefähr 1,7% relativ), bleibt robust gegenüber Oberflächendefekten und Bildgebungsstörungen und liefert Grad-CAM-Visualisierungen zur Interpretation relevanter Bildregionen.

Insgesamt liefert die Dissertation eine einheitliche, datengetriebene Pipeline zur Messung aus wissenschaftlichen Single-View-Videos. Der Ansatz vereinfacht die experimentelle Instrumentierung und ermöglicht eine präzise, automatisierte Schätzung der Tropfendynamik. Darüber hinaus erlaubt er es Forschenden, die Tropfenbreite kontinuierlich entlang

der gesamten Gleitstrecke zu überwachen und damit die Sichtfeldbegrenzungen der konventionellen Frontansichtsbildgebung zu überwinden. Über die Benetzung hinaus stellen die vorgeschlagenen Methoden breit einsetzbare Werkzeuge für lernbasierte Messverfahren unter praktischen Bildgebungsrestriktionen bereit.

Acknowledgment

First and foremost, I would like to express my sincere gratitude to my mentors and supervisors, Prof. Dr. Hans-Jürgen Butt, Prof. Dr. Michael Wand, Dr. Rüdiger Berger, and Dr. Oleksandra Kukharenko for their continuous guidance, valuable discussions, and encouragement throughout my doctoral research. Their expertise, critical insights, and scientific rigor have greatly shaped the quality and direction of this work.

I would also like to extend my special thanks to Prof. Dr. Yanhui Guo, Prof. Dr. Ulrich Schwanecke, Dr. Mahsa Salehi, and Dr. Navid Mohammadi Foumani for their invaluable discussions and support on Computer Vision and Time-Series related topics. Without their support, it would not have been possible to complete this journey with the quality it has.

Beyond scientific guidance, I owe every achievement to my parents, whose unwavering support and countless sacrifices have accompanied me through every step of my life. I am deeply grateful to my sister, who has brightened my world, and to my wife, who has shared every challenge and joy with patience and love, making this path truly meaningful.

I am sincerely thankful to Prof. Dr. Andrew Ng, Prof. Dr. Omid Akhavan, Dr. Mohsen Yazdinejad, Mr. Alireza Akhavanpour, and Dr. Majid Eyvazian, whose dedication and passion for science inspired my interest in physics and computer science, and from whom I have learned extensively.

Many aspects of our work are interconnected, and there are people who go beyond their formal responsibilities to support others with genuine kindness, dedication, and perfect timing. Their efforts make daily life easier for everyone. Therefore, I would like to express my sincere gratitude to Ms. Dominique Henz, Ms. Teresa Petry, Mr. Normen Mendez, Mr. Bouvisage, and Dr. Sarah Chagri for their exceptional support and commitment.

Contents

List of Figures

Chapter 1

Introduction

When designing and comparing surfaces, it is often necessary to quantify how a liquid interacts with a solid. This tendency of a surface to attract or repel a liquid is referred to as *wettability* and is commonly summarized by the *contact angle*, the angle formed where the liquid, solid, and surrounding fluid meet. Contact-angle measurements are therefore widely used as a practical tool for surface characterization. In equilibrium, the contact angle reflects interfacial energy balance (Young's equation), but in many applications the central requirement is drop mobility, for example, how easily a drop initiates motion and continues to slide. Importantly, a large contact angle does not guarantee low resistance to motion: some bio-inspired surfaces, such as rose-petal and Salvinia-type designs, exhibit high apparent contact angles while also showing strong lateral adhesion, meaning the drop can remain pinned and resist sliding even on a tilted surface. In such cases, contact-angle hysteresis (advancing minus receding angle) is often a more informative descriptor of pinning and sliding onset, yet hysteresis measured at the threshold of motion may not capture the evolving dynamics once the drop is moving. To obtain a more direct understanding of drop–surface interaction during motion, it is useful to quantify drop geometry, including width, because sliding and retentive forces depend on drop shape and contact-line conditions. Friction and retention forces are useful for detecting surface inhomogeneities, assessing interfacial stability, and monitoring viscoelastic energy dissipation, and they are also critical in anti-icing and surface coating quality. Several related studies have addressed these topics from different perspectives. For example, one study aimed to improve the measurement of receding contact angles and the characterization of drop retention on surfaces [1]. Other works used contact-angle and sliding-drop measurements to design and evaluate functional surfaces, for example to control sliding-drop charging [2], study drop-charging mechanisms on novel surfaces [3], and develop chemically robust superhydrophobic surfaces [4]. In the following sections, we introduce the core concepts of wetting and computational approaches that can help to extract drop geometry information and analyze it during sliding.

1.1 Wetting

1.1.1 Interfacial tension

Surface tension (γ) governs the shape of a liquid. It arises because molecules at the surface experience asymmetric cohesive forces from neighboring molecules, unlike those located within the bulk of the liquid (Figure 1.1a). Although it originates from molecular interactions, surface tension is considered a macroscopic quantity and is defined at the macroscopic level. Its units are either force per unit length ($\mathrm{N/m}$) or energy per unit area ($\mathrm{J/m^2}$). When expressed as energy per unit area, it is more commonly referred to as surface energy.

To reduce the total surface energy, a liquid adjusts its shape upon contacting a solid surface. When the solid surface energy (γ_S) exceeds the combined contributions of the liquid–air surface tension (γ_L) and the liquid–solid interfacial surface energy (γ_{SL}), i.e., $\gamma_S > \gamma_L + \gamma_{SL}$, the liquid spreads fully and completely wets the solid. Conversely, if the solid surface energy (γ_S) is less than the sum of the liquid–air and liquid–solid interfacial surface energies ($\gamma_S < \gamma_L + \gamma_{SL}$), the liquid only partially wets the surface, resulting in a droplet with a finite contact angle (Figure 1.1b). The contact angle is defined at the point where the liquid, solid, and air phases meet, known as the three-phase contact line. Consequently, interfacial surface energy plays a crucial role in quantifying wettability.

The connection between interfacial surface energies and the contact angle is quantitatively expressed by the Young equation [5].

$$\gamma_L \cos\theta_Y = \gamma_S - \gamma_{SL} \tag{1.1}$$

The Young equation can be interpreted from both thermodynamic and mechanical perspectives. From a thermodynamic standpoint, it describes the condition under which a system minimizes its total energy upon contact between a liquid and a solid. Mechanically, it accounts for the horizontal force balance of interfacial tensions at the three-phase contact line. The Young equation is strictly valid for equilibrium drops on ideal surfaces, where the contact angle is governed solely by interfacial tensions. In this context, a larger contact angle corresponds to a lower solid surface energy. Therefore, the contact angle described by the Young equation is also known as the intrinsic contact angle or Young's angle (θ_Y). In practice, the contact angle is not evaluated directly at the three-phase contact line. Instead, a goniometer estimates the angle by fitting the macroscopic drop profile over a finite region near the contact line, typically on length scales larger than 10 μm. As a result, the measured angle can deviate from the intrinsic Young angle θ_Y, especially on real surfaces where roughness and chemical heterogeneity affect the local interface. To distinguish the experimentally obtained macroscopic value from θ_Y, we denote it as the "apparent contact angle" θ_{app} (Figure 1.1c) [6].

1.1.2 Static contact angles

When dealing with real, non-ideal surfaces, various modified models have been developed to relate the intrinsic contact angle to the apparent contact angle. In cases where the surface exhibits physical roughness, the Wenzel model is commonly used

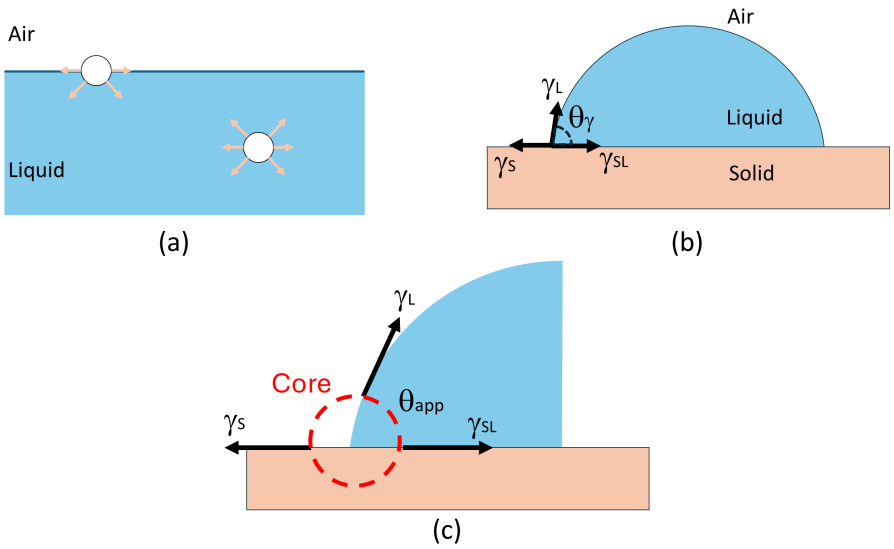


Figure 1.1: Schematic representations related to surface tension and contact angle. (a) Molecular origin of surface tension: molecules at the liquid–air interface experience unbalanced cohesive forces compared to those in the bulk. (b) The Young equation at the three-phase contact line, showing the balance of interfacial tensions (γ_S, γ_L, and γ_{SL}) and the intrinsic contact angle (θ_Y). (c) Apparent contact angle (θ_{app}) measured at the macroscopic scale, including the definition of the "core" region near the contact line where microscopic effects dominate.

to estimate the deviation of the apparent contact angle from the intrinsic one (Figure 1.2a) [7]:

$$\cos\theta_{\text{app}} = r\cos\theta_Y \tag{1.2}$$

The parameter r represents the roughness of the solid surface. For rough surfaces ($r > 1$), if the intrinsic contact angle satisfies $\theta_Y > 90°$, the apparent contact angle becomes larger than the intrinsic one ($\theta_{\text{app}} > \theta_Y$). Conversely, when $\theta_Y < 90°$, the apparent contact angle decreases ($\theta_{\text{app}} < \theta_Y$). In essence, surface roughness amplifies the inherent wetting behavior, whether the surface is hydrophobic or hydrophilic. However, a limitation of the Wenzel model is that for sufficiently large r, the predicted apparent contact angle may exceed $180°$ or fall below $0°$, which is physically unrealistic. Therefore, the model is only valid within a limited range of surface roughness [8].

Beyond physical roughness, real surfaces may also exhibit chemical heterogeneity. For instance, a perfectly smooth surface composed of two distinct materials, each with its own intrinsic contact angle (θ_{Y1} and θ_{Y2}), represents such a case. If the characteristic size of these individual regions is much smaller than that of the drop, the apparent contact angle on this chemically heterogeneous surface can be estimated by the Cassie relation, Eq. (1.3):

$$\cos\theta_{\text{app}} = \phi_1\cos\theta_{Y1} + \phi_2\cos\theta_{Y2} \tag{1.3}$$

Here, ϕ_1 and ϕ_2 represent the fractional surface areas occupied by the two different components, with the constraint $\phi_1 + \phi_2 = 1$. When one of the two components

is air, for which the intrinsic contact angle is 180°, Eq. (1.3) reduces to the Cassie–Baxter equation [9] (Figure 1.2b):

$$\cos\theta_{\text{app}} = \phi\cos\theta_Y - (1-\phi) \tag{1.4}$$

Equations (1.3) and (1.4) are macroscopic relations and assume that the wetting state is uniform at the scale of the three-phase contact line, so that the contact line effectively samples a representative solid fraction (and, in the Cassie–Baxter state, a representative solid–air composite interface). If the surface consists of large zones with different textures or chemistries, or if the Cassie/Wenzel state varies spatially (e.g., partial impalement near the contact line), the measured apparent angle can be dominated by the local region at the contact line and may deviate from the simple area-fraction prediction.

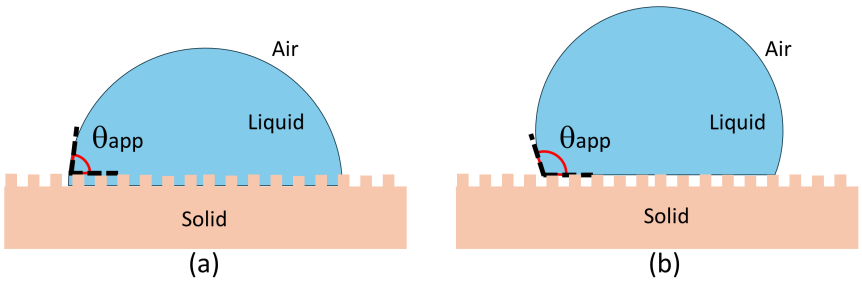


Figure 1.2: Illustration of the Wenzel and Cassie–Baxter wetting regimes. In the Wenzel regime, the liquid fully follows the surface texture, increasing contact with the solid. In the Cassie–Baxter regime, air remains trapped beneath the droplet, resulting in a composite interface and reduced solid–liquid contact.

1.1.3 Contact angle hysteresis

All of the aforementioned models are grounded in the Young equation and therefore assume an equilibrium contact angle on an ideal surface. In practice, however, real surfaces exhibit roughness and chemical heterogeneity, which can pin the three-phase contact line and lead to multiple metastable configurations. As a result, even an apparently static drop may not adopt the unique Young angle; instead, its measured contact angle typically lies between the quasi-static advancing angle (θ_A) and the quasi-static receding angle (θ_R).

When a drop begins to move, the contact angle at the leading edge is referred to as the advancing contact angle, while the angle at the trailing edge is the receding contact angle. The difference between these two angles is known as contact angle hysteresis.

Methods for measuring advancing and receding contact angles include the Wilhelmy-plate method [10], the pulling method [11, 12], the inflated/deflated drop method [13, 14], and the tilted-plate method [15, 16](Figure 1.3). These methods differ not only in how the contact line is mobilized but also in how the measurement is taken across the surface.

In the Wilhelmy-plate method, a thin plate of well-defined geometry is vertically immersed into, or withdrawn from, a liquid bath while the wetting force is recorded

with a tensiometer [17, 18]. In general, the capillary force acting on the probe can be written as

$$F = \gamma P \cos\theta_{a/r}, \tag{1.5}$$

where γ is the surface tension of the liquid, P is the perimeter of the three-phase contact line on the probe, and $\theta_{a/r}$ is the advancing/receding contact angle depending on whether the probe is advanced or withdrawn [18]. For a thin plate, $P \approx 2w$, where w is the *width* of the plate [17].

The Wilhelmy-plate method is simple and does not require empirical correction factors, but reliable results require a clean probe and careful avoidance of contamination [17]. Because the measured force integrates along the entire three-phase contact line on the probe, the Wilhelmy-plate method yields a line-averaged (effective) contact angle and cannot resolve local variations caused by chemical heterogeneity or isolated surface defects.

The remaining three methods estimate contact angles based on drop shape analysis, making their accuracy dependent on camera resolution, image quality, the chosen fitting model, and the image processing algorithms applied.

In the pulling method, a drop is anchored with a spring or flexible element and dragged across the surface by a motorized stage. This setup allows the contact-line velocity to be controlled by the motor. This approach measures the contact angle along a single sliding path.

In the inflated/deflated drop method, the contact line is mobilized by changing the drop volume through liquid injection or withdrawal with a syringe. The contact-line velocity is directly controlled by the flow rate. The syringe's position within the drop significantly influences the results. To minimize this effect, the needle is typically placed near the center of the drop and close to the solid–liquid interface. This method analyzes only a small spot of the surface for one measurement.

In the tilted-plate method, a drop is placed on an inclined surface and allowed to slide under gravity. In contrast to force-controlled methods, the contact-line speed is not imposed independently but results from the balance between the driving component of gravity and dissipation/pinning, and therefore depends on the tilt angle, drop volume, liquid viscosity, and contact-angle hysteresis. Consequently, the measured angles may reflect dynamic effects unless the motion is sufficiently slow (quasi-static).

Once the advancing and receding contact angles are known, the contact-angle hysteresis follows directly as $\Delta\theta = \theta_a - \theta_r$. Contact angle hysteresis plays a significant role in daily phenomena, as it governs the friction experienced by liquid drops on surfaces. Specifically, it determines the lateral adhesion force required to initiate droplet motion. This force can be estimated using the Furmidge equation [19, 20]:

$$F = kw\gamma_L(\cos\theta_r - \cos\theta_a) \tag{1.6}$$

Here, F is the lateral adhesion force, w is the width of the drop's contact area, and $k \approx 1$ is a geometric factor that depends on the drop shape [21, 22].

1.1.4 Roll-off angle

A related metric in the tilted-plate method is the roll-off (or sliding) angle, α_{ro}, defined as the smallest inclination angle at which a sessile drop first starts to move (i.e.,

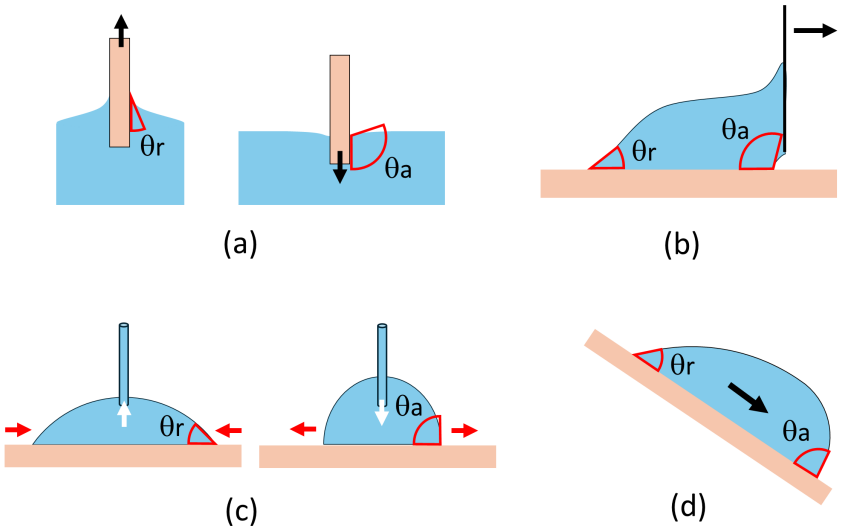


Figure 1.3: Schematic illustration of methods for measuring advancing (θ_a) and receding (θ_r) contact angles, which define contact angle hysteresis. (a) Wilhelmy-plate method where a surface is immersed into or withdrawn from a liquid. (b) Pulling a drop horizontally using a solid boundary to distinguish the front (advancing) and rear (receding) contact angles. (c) Controlled volume increase or decrease by injecting or withdrawing liquid through a needle. (d) Sliding a drop down an inclined plane to observe dynamic advancing and receding angles.

undergoes sustained sliding/rolling) under gravity when the tilt is increased quasi-statically [23]. Based on Eq. (1.6), the roll-off angle α_{ro} can be estimated by balancing the driving component of gravity, $mg\sin\alpha_{\mathrm{ro}}$, with the lateral adhesion force, yielding

$$\sin\alpha_{\mathrm{ro}} = \frac{k\,w\,\gamma_L}{mg}\,(\cos\theta_r - \cos\theta_a), \tag{1.7}$$

Here, $g = 9.81\,\mathrm{m/s^2}$ is the gravitational acceleration, m is the mass of the drop. The mass is related to the drop volume V by $m = \rho V$, and ρ is the liquid density. Since the contact width w typically scales with the drop radius, the right-hand side of Equation 1.7 scales approximately with $V^{-2/3}$. As a result, the sliding angle α, also referred to as the roll-off angle, depends on the volume of the drop.

Based on Equation 1.7, in the absence of contact angle hysteresis, drops would begin to slide on inclined surfaces even at very small tilt angles. While such low adhesion could be beneficial for applications like keeping glass surfaces or car windshields clean, preventing fogging, or efficiently removing condensation drops in heat exchangers, it would pose serious challenges in other contexts. For instance, precise drop control is crucial in processes such as printing, painting, coating, and the uniform application of herbicides or insecticides. Therefore, contact angle hysteresis plays a vital role in many practical and industrial applications and is fundamentally important in everyday life.

Therefore, knowledge of the lateral adhesion force makes it possible to answer several practical questions: At what tilt angle will a drop begin to slide? How much external force is required to displace a drop from a given surface? What degree of surface modification is necessary to achieve precise control over drop mobility?

Addressing these questions is essential for the rational design of surfaces in applications such as microfluidic systems, self-cleaning coatings, and liquid-repellent technologies.

While the roll-off angle characterizes the onset of motion in a quasi-static sense, many applications require contact angles during sustained sliding, i.e., dynamic advancing and receding angles.

1.1.5 Dynamic contact angles

Once sliding is initiated in the tilted-plate method, it becomes challenging to maintain well-defined measurement conditions and to capture the evolving drop shape and contact angles reliably. As a result, in many practical implementations, the tilted-plate method is used mainly to determine the onset of motion (roll-off angle) and quasi-static advancing/receding angles. To capture dynamic contact angles, the drop must be imaged while sliding, which presents greater technical challenges than recording stationary sessile drops [24, 25]. Analyzing high-speed videos of sliding drops presents significant challenges, primarily due to the limited resolution in the region near the drop interface. Measuring contact angles under such conditions is further complicated by the deformable and often asymmetric shape of the drop, along with the fact that dynamic θ_a and θ_r can vary independently. These difficulties are especially pronounced in the presence of surface defects, which locally modify the drop–substrate interaction. To reduce the reliance on high-speed cameras and simplify image processing, studies often employ high-viscosity liquids to slow down drop motion [23]. However, several challenges still persist. To address the difficulties in measuring dynamic contact angles, some works have introduced numerical techniques such as the double-sided elliptical fitting method [26].

Dynamic contact angles are crucial whenever the contact line moves, since they directly influence dissipation, interfacial friction, and the onset and evolution of sliding. They therefore matter in applications such as coating and printing, and drop transport and manipulation in microfluidic systems. In this dissertation, dynamic advancing and receding contact angles during sliding are a central target quantity. We develop methods in the following chapters to measure them accurately and robustly from side-view video data on real, non-ideal surfaces, and we subsequently use the extracted angles for different purposes depending on the specific analysis objective.

1.1.6 Viscous forces

Viscous forces represent the internal friction within a fluid, opposing motion when adjacent liquid layers move past each other under shear. Their magnitude scales with the dynamic viscosity η of the liquid and the contact line velocity U, following $F \propto \eta U$. Numerical simulations have shown that at low sliding velocities, viscous dissipation is dominated by the wedge region near the contact line, whereas at higher velocities the contributions from the wedge and bulk regions become comparable [27].

A useful way to quantify the interplay between viscous forces and surface tension

is the capillary number, defined as

$$\mathrm{Ca} = \frac{\eta U}{\gamma}$$

where η is the dynamic viscosity, U the characteristic (e.g., contact line) velocity, and γ the liquid–gas surface tension. The capillary number, Ca, is a dimensionless number that measures the relative magnitude of viscous forces compared with interfacial (capillary) forces [28].

At low capillary number, $Ca \lesssim 10^{-3}$, capillary forces dominate over viscous stresses. A strictly static drop corresponds to $U = 0$ and therefore $Ca = 0$. On an ideal, chemically homogeneous and smooth surface, the contact angle then equals the Young angle θ_Y. On real surfaces, however, contact-line pinning can lead to metastable static states.

As Ca increases, viscous dissipation near the moving contact line becomes important and the contact angles become velocity dependent. We denote these dynamic angles by $\theta_a(U)$ and $\theta_r(U)$, which generally deviate from their quasi-static limits θ_A and θ_R; typically, $\theta_a(U)$ increases with U while $\theta_r(U)$ decreases with U. The whole drop shape departs from equilibrium. This explains why the dynamic advancing and receding angles can span a wide range and vary almost independently, making accurate contact angle determination difficult.

At high capillary number, $Ca \gtrsim 10^{-2}$, viscous forces dominate. Deformation becomes strong and persistent. Beyond a critical Ca, viscous stress can overcome capillarity and produce breakup in confined flows, as shown for droplets in tapered microchannels [29].

1.1.7 Friction force

The friction force experienced by a moving drop arises from multiple mechanisms, including hydrodynamic viscous dissipation in both the bulk and the wedge region [24]; contact-line friction resulting from the thermally activated motion of liquid molecules near the contact line [30]; pinning and depinning effects caused by surface heterogeneities [31]; elastocapillary deformation when interacting with soft substrates [32]; and electrostatic retardation effects associated with slide-induced electrification [33]. It is evident that knowledge of the friction force is not only important for fundamental studies but also has numerous practical applications. These include detecting surface inhomogeneities, evaluating interfacial stability, and monitoring viscoelastic energy dissipation [34]. Moreover, friction force plays a critical role in technologies such as anti-icing [35] and assessing the quality of surface coatings [36].

A recent study by Li et al. investigated the behavior of drops sliding down inclined surfaces and proposed an empirical relation describing the friction force F_f as a function of drop velocity U [33]:

$$F_f = F_0 + \beta \omega U \eta \tag{1.8}$$

where β is a dimensionless friction coefficient, ω is the width of the drop during sliding, η is the liquid viscosity, and F_0 represents the friction force extrapolated to zero velocity. To establish this relation, they measured the sliding velocity, contact

width, contact length, and both advancing and receding contact angles of drops moving on inclined flat surfaces made of various materials. Also, to obtain the drop width during motion, two mirrors were added to the setup to enable front-view observation. However, this approach limited the measurable field of view to approximately 1.5 cm and introduced additional complexity to the experimental arrangement.

As previously mentioned, the empirical formulation requires several features extracted from sliding drops, such as the advancing and receding contact angles, as well as the drop length. However, analyzing videos of drops sliding on a tilted plate is challenging due to the low resolution of the region where the drop appears. Drops typically occupy only a small portion of each frame, suffer from limited image detail, and lose symmetry as their contact angles evolve during motion. These factors render conventional methods—such as circle or ellipse fitting, which rely on fixed geometric assumptions—ineffective for accurate shape analysis.

Another critical and technically demanding feature to measure is the drop width, which poses even greater challenges than length or contact angles. Drop width data can be obtained through either bottom-view or front-view imaging of sliding drops. Bottom-view imaging, however, is limited to transparent substrates. Front-view imaging over a sliding distance of approximately 1.5 cm can be achieved by adding a second, time-synchronized high-speed camera [11]. Alternatively, this second camera can be avoided by placing two mirrors at the beginning and end of the drop's sliding path [33]. One mirror reflects light from a source positioned behind the drop, while the other enables front-view video capture.

The main experimental difficulty in both setups lies in achieving optimal lighting conditions for the additional optical components. Positioning mirrors along the optical path is delicate and may lead to reduced image contrast due to secondary reflections. If a second camera is used, it requires a dedicated front-view illuminator, effectively doubling the equipment on a platform that must be capable of rotating by 90°. More importantly, in both approaches, the drop moves toward the camera or mirrors, which limits the effective field of view for front-view imaging to roughly 1.5 cm. This severely constrains the observable range along the sliding direction.

1.2 Image-based contact angle estimation

1.2.1 Asymmetric problem

Axisymmetric Drop Shape Analysis (ADSA) determines the contact angle by fitting the drop profile to the Young–Laplace equation under the assumption of perfect axisymmetry. It provides high accuracy across a wide range of contact angles and can also extract parameters such as surface tension. Several variants exist, including axisymmetric drop shape analysis—diameter (ADSA-D), axisymmetric drop shape analysis—profile (ADSA-P), axisymmetric drop shape analysis—height and diameter (ADSA-HD), and Low-Bond ADSA (LB-ADSA) [37, 38]. While these methods are highly precise, they remain limited to axisymmetric scenarios such as sessile drops. For more complex cases, including sliding or bouncing drops, or situations involving asymmetric deformation, more flexible methods are required.

1.2.2 Tangent fitting

Contact angle measurement often begins with the most straightforward approach: drawing a baseline along the substrate and a tangent to the drop profile at the three-phase contact line, then determining the angle between them. This basic goniometric or "tangent" method is fast, intuitive, and performs reasonably well for moderate wettability. It also allows independent reporting of left and right contact angles when the drop is asymmetric [39]. The method requires minimal preprocessing and little modeling effort. However, it has notable limitations: it depends on a clean, well-defined drop edge and is highly sensitive to image noise, as it relies on only a few pixels near the contact point. Baseline leveling and the exact localization of the contact point strongly influence the result, and even small image rotations or motion blur can cause errors of several degrees. It is therefore evident that numerous approaches have been developed to enable reliable contact angle measurement under real experimental conditions.

1.2.3 Geometric primitive fitting

Circle and ellipse fitting methods approximate a drop's profile by simple geometric curves, based on the assumption that a sessile drop resembles a segment of a sphere or an ellipsoid. The contact angle is obtained from the slope of the fitted curve at the contact point. Typically, the fit is performed on the upper portion of the drop profile, away from the baseline, and then extrapolated to the baseline to determine the contact angle. Computationally, a nonlinear least-squares algorithm, such as the Levenberg–Marquardt method, is often employed to obtain the best-fit parameters. Circle fitting performs well for small to moderate contact angles where gravitational effects are negligible (low Bond number), and the drop profile remains close to a spherical cap. However, the method becomes unreliable for highly hydrophobic drops; beyond a certain contact angle, the spherical-cap approximation fails and large errors can occur [40].

Ellipse-fitting methods approximate the drop profile with a general conic section, offering more degrees of freedom than circle fitting by allowing different curvatures in the vertical and horizontal directions. Recently, a method has been introduced that fits a full ellipse without assuming alignment with the substrate [41]. In this approach, the ellipse axes need not be horizontal or vertical, enabling a tilted ellipse fit. This capability is useful for non-axisymmetric drops where the baseline is not clearly visible, such as overlapping droplets in condensation experiments where the actual contact line is obscured. Furthermore, a double-sided elliptical fitting technique has been proposed to determine advancing and receding angles simultaneously by fitting separate ellipses to each side of a drop profile [26]. Ellipse fitting generally performs better for larger contact angles, where circle fitting would systematically underestimate the curvature. While ellipse fitting adds flexibility, it still assumes the profile is a smooth conic section; in reality, drops under gravity follow a catenary-like Young–Laplace curve rather than a perfect ellipse. Moreover, purely geometric fits may not pass exactly through the contact point, even though the contact angle should be evaluated there.

1.2.4 Polynomial and spline fitting

Rather than imposing a simple geometric shape, many methods fit a polynomial curve (or a spline) to the drop profile. The idea is to use a generic smooth curve that approximates the true outline, then compute the contact angle from the slope of that curve at the contact point. A common approach is to select a short segment of the edge near the contact point and fit a polynomial,

$$y(x) = a_0 + a_1 x + a_2 x^2 + \cdots + a_n x^n, \tag{1.9}$$

typically in image–plane coordinates. The contact angle is then

$$\theta = \arctan\left(\left. \frac{dy}{dx} \right|_{\text{contact}} \right), \tag{1.10}$$

evaluated at the baseline. This method is popular because it uses more information than a single tangent and remains computationally simple. It does not assume a particular physical model, only that the outline is smooth and locally well approximated by a polynomial [42].

Early implementations required the user to choose both the polynomial order and the data range to be fitted, which introduced subjectivity and variability. In addition, many implementations locate the baseline via the drop reflection in static sessile–drop setups, a cue that is often inaccessible in scenarios such as sliding drops. A central limitation of polynomial fitting is precisely this choice of order and number of points: different selections can yield different contact angles and thus add uncertainty [42, 43]. Atefi et al. addressed this with a “differentiator mask” algorithm that systematically varies polynomial order and dataset size, identifying plateaus in the contact–angle–versus–point–count curve where the angle remains stable. The optimal order is chosen from this plateau, avoiding overfitting and underfitting. Combined with subpixel edge–detection refinement, the method achieved errors of less than 0.4° for $\theta < 60^\circ$ and less than 1° for $40^\circ \leq \theta \leq 170^\circ$ using high–resolution images. For steep profiles, they employed a polynomial–in–polar–coordinates approach, transforming the profile into (r, θ) space about the drop apex to linearize high–slope regions. This polar–coordinate method requires the apex to be visible, and thus cannot be applied to geometries such as needle–in–drop configurations, but for full–drop profiles it substantially improved accuracy and consistency. High–resolution images and subpixel edge–detection are essential for polynomial fitting because they minimize discretization and localization errors at the contact point, which can otherwise cause large deviations in the computed contact angle.

Spline fits model the drop edge using piecewise polynomials. In DropSnake, a B-spline active contour is initialized around the drop and deforms toward the silhouette by minimizing an energy functional combining image-gradient attraction and smoothness constraints [42]. Control points are placed more densely near the contact line to capture local geometry, and the contact angle is obtained from the tangent to the spline at that point. This approach incorporates global shape information while preserving local accuracy, and it avoids the need to select a specific polynomial order. However, when advancing or receding angles vary sharply over small regions—as in sliding drops encountering surface defects or heterogeneities—the global nature of the spline can smooth out such local variations. Because the contour is constrained by overall continuity and smoothness, this may bias the tangent

measurement near the contact line, particularly when spatial resolution is limited or motion blur is present.

1.3 Sequence models

1.3.1 Extrinsic time series regression

Time Series Extrinsic Regression (TSER) is the task of learning a function that maps a complete time series to a single continuous value, rather than to a class label or to a future sample. In statistics, this is treated as scalar-on-function regression (SoFR), where a time series is viewed as a function and the goal is to predict a scalar response from that function [44].

TSER matters because many targets are numeric summaries that depend on patterns across an entire sequence rather than on the most recent points. Examples include estimating heart rate or respiratory rate from ECG or PPG [45], predicting live fuel moisture content (LFMC) from year-long satellite series, and forecasting agricultural outcomes such as crop yield from spectral-temporal patterns [46].

Despite its importance, TSER tools lag behind those for classification and forecasting. Tabular regressors ignore temporal order, and domain-specific solutions often fail to generalize. Forecasting typically assumes that near-future values resemble recent history, using methods such as ARIMA or exponential smoothing [47]. This locality assumption makes forecasting unsuitable when the goal is to summarise an entire sequence.

1.3.2 Vanilla recurrent neural networks

In many fields of research and practical applications, information arrives in a sequence. Each part of the sequence is shaped by what came before it. These time-based links can be as important as the data itself. They show up when spoken words form a sentence. They appear when market values rise and fall in a pattern. They can be seen in the repeating signals of time-series data. They also appear in the way scenes flow together in a video. Standard feedforward neural networks process each input on its own. They do not pass information forward from one step to the next. As a result, they cannot capture how earlier events affect later ones. Recurrent neural networks (RNNs) were developed to solve this problem [48, 49]. They keep a hidden state that changes over time. This hidden state carries past information forward so that earlier steps can influence later predictions. By doing so, RNNs can model data where order and timing matter.

Recurrent neural networks extend conventional feedforward architectures by introducing recurrent connections, allowing the network to pass information from one time step to the next. Given an input sequence $\{x_1, x_2, \ldots, x_T\}$, the hidden state h_t at time t is computed as:

$$h_t = \tanh(W_{xh}x_t + W_{hh}h_{t-1} + b_h) \tag{1.11}$$

$$y_t = W_{hy}h_t + b_y \tag{1.12}$$

where W_{xh}, W_{hh}, and W_{hy} are trainable weight matrices, b_h and b_y are bias terms, and $\tanh$ is the hyperbolic tangent non-linear activation [48, 50].

RNNs have been successfully applied in tasks such as language modeling [49], handwriting recognition [51], and sequence prediction in physical systems [52]. Their ability to capture short-term temporal patterns, combined with their relatively small number of parameters, makes them suitable for real-time and resource-constrained applications. However, their sequential nature hinders parallelization, and the repeated transformations during backpropagation through time cause gradients to either vanish or explode, making the learning of long-term dependencies difficult [53, 54, 55]. These limitations provided the impetus for more advanced recurrent architectures that could stabilize training and extend memory capacity.

1.3.3 Long short-term memory

RNNs were an important step forward in sequence modeling. They allowed models to use information from earlier steps in a sequence. But they also brought new problems. When the sequence becomes long, training can become unstable. Gradients can shrink toward zero or grow too large. This makes it hard for the network to learn connections between events far apart in time [53, 54]. To address this, the long short-term memory (LSTM) architecture was developed [56]. LSTM networks extend the idea of recurrent processing by introducing a structured gating mechanism that carefully regulates the flow of information. Instead of simply passing the hidden state forward, the LSTM maintains a dedicated memory cell c_t that can preserve information over long time spans. At each time step, three gates determine how the memory is updated. The forget gate f_t decides which parts of the existing memory should be discarded. The input gate i_t controls which new information should be added to the memory. Finally, the output gate o_t determines which parts of the memory should influence the hidden state and, ultimately, the output. By adjusting these gates dynamically, the LSTM can retain long-term dependencies without suffering from the gradient vanishing or explosion issues seen in vanilla RNNs.

The LSTM computations at time step t are:

$$f_t = \sigma(W_f x_t + U_f h_{t-1} + b_f) \tag{1.13}$$

$$i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i) \tag{1.14}$$

$$\tilde{c}_t = \tanh(W_c x_t + U_c h_{t-1} + b_c) \tag{1.15}$$

$$c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c}_t \tag{1.16}$$

$$o_t = \sigma(W_o x_t + U_o h_{t-1} + b_o) \tag{1.17}$$

$$h_t = o_t \odot \tanh(c_t) \tag{1.18}$$

where $\sigma(\cdot)$ is the logistic sigmoid function, $\odot$ denotes element-wise multiplication, W_* and U_* are weight matrices, and b_* are bias terms.

The gating approach allows the network to selectively remember important information, ignore irrelevant details, and control how much past context influences the present output. This capability is particularly useful in applications that require modeling both short- and long-term dependencies, such as speech recognition [57], machine translation [58], and spatiotemporal video analysis [59].

In datasets with limited size, the high capacity of the model can lead to overfitting. Regularization can be achieved in several ways, including choosing an appropriate model capacity (e.g., fewer layers or a smaller hidden dimension) and applying training-time techniques. Commonly used approaches for LSTMs include dropout, which randomly disables a fraction of units during training to reduce co-adaptation [60], and weight decay, which penalizes large weights and often improves generalization in practice [50]. We note that the effect of weight decay can depend on architectural components and the overall training setup.

1.3.4 Gated recurrent unit

One trade-off of LSTMs is their computational cost. The additional gates and parameters increase training time and memory usage compared to simpler recurrent architectures. This complexity can be a limiting factor in real-time or resource-constrained environments, and it has motivated the development of more lightweight alternatives such as the gated recurrent unit (GRU) [61].

The GRU simplifies the LSTM design while preserving its ability to capture long-term dependencies. It removes the explicit memory cell and merges it with the hidden state. Only two gates are used: the update gate z_t and the reset gate r_t. The update gate determines how much of the previous hidden state should be carried forward to the current time step, acting as a combined mechanism for both remembering and forgetting. The reset gate controls how the new input is combined with the past hidden state, allowing the network to drop irrelevant historical information when producing the candidate hidden state.

The GRU computations at time step t are:

$$z_t = \sigma(W_z x_t + U_z h_{t-1} + b_z) \tag{1.19}$$

$$r_t = \sigma(W_r x_t + U_r h_{t-1} + b_r) \tag{1.20}$$

$$\tilde{h}_t = \tanh(W_h x_t + U_h(r_t \odot h_{t-1}) + b_h) \tag{1.21}$$

$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \tilde{h}_t \tag{1.22}$$

By using fewer gates and parameters than the LSTM, the GRU often trains faster and uses less memory, making it attractive for large-scale applications and deployment on devices with limited resources. Empirical studies have shown that GRUs can achieve performance comparable to LSTMs in many tasks, including sequence modeling [62], sentiment analysis [63], and anomaly detection in time series [64]. In some cases GRUs even outperform LSTMs, especially in situations where the dataset is smaller or when the training budget (time/compute) is limited [65].

While the absence of a separate cell state makes GRUs simpler, it can also limit their ability to model extremely long-term dependencies in certain domains. The choice between LSTM and GRU therefore depends on the task requirements, data size, and computational constraints, and is often determined empirically.

Bidirectional recurrent networks are a standard extension of RNNs, LSTMs, and GRUs in which two recurrent passes are run over the same sequence, one forward and one backward, and their representations are combined [66]. This can be beneficial in offline settings where the full sequence is available, because the model can use both earlier and later frames to infer quantities that are noisy or only weakly

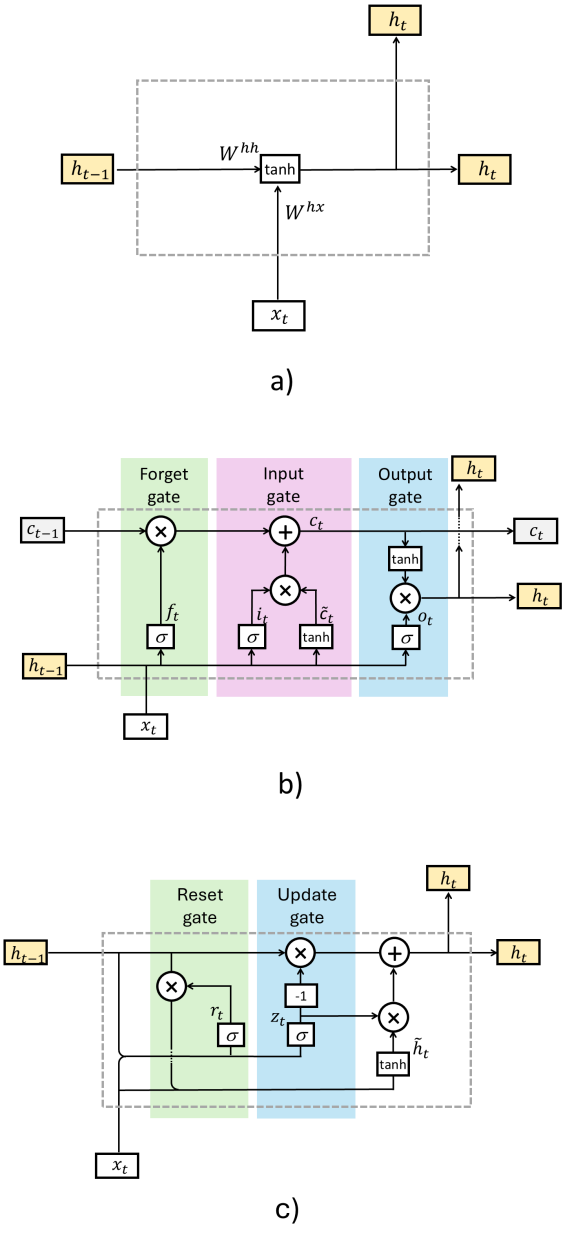


Figure 1.4: Architectural comparison of recurrent neural network variants: (a) a vanilla RNN, where the hidden state h_t is updated using the current input x_t and the previous hidden state h_{t-1}; (b) a long short-term memory (LSTM) unit, which incorporates a cell state c_t and three gates (forget, input, and output) to regulate information flow and preserve long-term dependencies; (c) a gated recurrent unit (GRU), which simplifies the LSTM design by combining the cell and hidden state, and using two gates (reset and update) to control information retention and update.

observable at individual time steps [67, 50]. In this thesis, bidirectional variants are therefore included where appropriate (e.g., as part of the model comparisons in the multivariate sequence analysis study). The underlying gating and training considerations are the same as for the unidirectional case.

1.3.5 Transformers

Recurrent models handle sequences step by step. This limits parallelization and makes it hard to link distant points in time. Transformers [68] replace recurrence with self-attention. Instead of passing information through a chain of hidden states, self-attention lets each time step connect directly to every other. This removes the bottleneck of sequential processing and makes long-range patterns easier to learn.

A Transformer encoder first maps each input $\{x_1, x_2, \ldots, x_T\}$ to an embedding $\{e_1, e_2, \ldots, e_T\}$. Positional encodings are added so the model knows the order of steps. For each position, the model computes queries Q, keys K, and values V. Attention weights are:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right)V \tag{1.23}$$

The softmax scores show how strongly each time step attends to others. Multi-head attention repeats this process with different learned projections, then combines the results. Each head can focus on different temporal patterns.

Transformers have been adapted for time series in many ways. The Time Series Transformer [69] applied the encoder directly for classification and regression. Informer [70] used sparse attention to handle very long sequences. The Temporal Fusion Transformer (TFT) [71] mixed attention with gating and static covariate encoders. PatchTST [72] grouped consecutive points into patches before attention, improving efficiency.

For time series, Transformers bring clear benefits. They capture relationships between all time steps in one layer. They train in parallel. They model both local and global patterns. But they have costs. Standard attention needs memory and compute that grow with the square of sequence length. Efficient attention variants [73, 74] aim to address this by reducing attention complexity. They also lack the natural locality bias of RNNs and CNNs. Without this built-in bias toward local structure, Transformers must learn locality from the data, which often requires larger datasets to achieve good performance, especially when short-term patterns are important [75]. One solution is to combine Convolutional layers with Transformers. CNNs extract local features efficiently, focusing on short-term patterns and reducing sequence length through pooling or striding. This lowers the computational load for the Transformer and provides richer, noise-filtered inputs. The Transformer then models the long-range dependencies that CNNs cannot capture well.

In the original Transformer [68], the architecture has no inherent notion of order because self-attention is permutation-invariant. To make the model aware of the position of each element in a sequence, positional encodings are added to the input embeddings before they enter the encoder. Vaswani et al. proposed a deterministic sinusoidal Absolute Position Encoding (APE), where each position index i is mapped to a vector of the same dimension as the embeddings. Even dimensions

use sine functions and odd dimensions use cosine functions, with frequencies that vary across dimensions according to an exponential scale. The formulation is

$$\omega_k = 10000^{-2k/d_{\text{model}}} \tag{1.24}$$

$$p_i(2k) = \sin(i\,\omega_k), \quad p_i(2k{+}1) = \cos(i\,\omega_k), \tag{1.25}$$

where d_{model} is the embedding dimension and $k \in \{0, 1, \ldots, \frac{d_{\text{model}}}{2} - 1\}$ is the dimension index. This encoding scheme has several desirable properties: it enables the model to attend by relative positions because any offset between positions produces a consistent phase difference in the encoding, and it allows extrapolation to sequence lengths longer than those seen during training.

While sinusoidal APE works well in high-dimensional NLP settings, it is less effective for short time series with low-dimensional embeddings, which are common in multivariate time series data. At low dimensions, the dot product between positional embeddings does not reliably decrease with increasing positional distance, and different positions can end up with very similar encoding vectors. This weakens the model's ability to distinguish nearby from distant time steps and reduces its awareness of temporal proximity.

To address these issues, ConvTran [76] introduced the time Absolute Position Encoding (tAPE), which adapts the sinusoidal formulation to the specific sequence length L and embedding dimension d_{model}. The scaling constant in the frequency term is replaced by a ratio that incorporates d_{model}/L, ensuring that the positional frequencies are better matched to the resolution of the time series. The modified encoding is

$$\omega_k^{\text{tAPE}} = \omega_k \cdot \frac{d_{\text{model}}}{L}, \tag{1.26}$$

$$p_i^{\text{tAPE}}(2k) = \sin(i\,\omega_k^{\text{tAPE}}), \quad p_i^{\text{tAPE}}(2k{+}1) = \cos(i\,\omega_k^{\text{tAPE}}) \tag{1.27}$$

When $d_{\text{model}} = L$, the tAPE formulation reduces exactly to the original APE. This adjustment preserves distance awareness and isotropy even for low-dimensional embeddings, improving the Transformer's ability to represent both short- and long-range temporal dependencies in time series data.

Several works follow the design of combining CNNs with Transformers and improving positional encoding for time series. For multivariate forecasting, STCTN couples convolutional attention with Transformer layers to capture local and global temporal dependencies [77]. Transformer-enhanced Temporal Convolutional Networks integrate a TCN front end with attention to model short- and long-range periodicity [78]. In time-series classification, ConvTran [76] adopts a CNN–Transformer structure with temporal and spatial convolutions and introduces two position encodings (tAPE and eRPE) to improve performance.

1.4 Vision models

1.4.1 Convolutional neural networks

Convolutional neural networks (CNNs) introduce locality and translation equivariance through spatial weight sharing, which makes them data efficient and well

matched to image statistics [79]. In practice, CNN layers implement cross-correlation rather than true convolution, omitting the kernel flip used in classical signal processing; this simplification has no effect on representational power since the kernels are learned.

A 2D convolutional layer maps an input tensor $x \in \mathbb{R}^{C\times H\times W}$ (with C input channels and spatial dimensions $H \times W$) to output feature maps $y \in \mathbb{R}^{C'\times H'\times W'}$ (with C' output channels and possibly reduced spatial dimensions) by

$$y_{c',i,j} = b_{c'} + \sum_{c=1}^{C}\sum_{u=0}^{k_h-1}\sum_{v=0}^{k_w-1} W_{c',c,u,v}\, x_{c,i+u,j+v} \tag{1.28}$$

where $W_{c',c,u,v}$ are the learnable kernel weights of spatial size $k_h \times k_w$, $b_{c'}$ is a bias term for each output channel, (i, j) indexes the spatial location in the output, and (u, v) indexes positions within the kernel. This operation aggregates local patches from all C input channels into each output channel using the same weights across all spatial locations, which greatly reduces the number of parameters compared to fully connected layers.

The convolution output is typically passed through a pointwise nonlinearity ϕ (e.g., ReLU) to introduce non-linear feature mappings, and may be followed by spatial downsampling. Downsampling can be achieved by striding (moving the convolution window by steps greater than one) or by pooling—a separate operation that replaces a small region by a single summary value, such as the maximum (max pooling) or the average (average pooling). Pooling reduces spatial resolution, increases the receptive field, and provides a degree of invariance to small translations.

Stacking multiple convolutional layers expands the effective receptive field, allowing deeper layers to capture increasingly global patterns while keeping parameters modest. Batch normalization [80] further stabilizes training by normalizing intermediate activations, improving convergence.

CNNs have served as the backbone for many breakthrough vision systems: AlexNet established large-scale image classification with deep convolutional stacks [81], VGG clarified the benefits of depth with small kernels [82], Inception improved compute–accuracy tradeoffs via multi-branch filters [83], and ResNets set the modern baseline through residual learning [84]. For many applications such as detection and segmentation, region-based CNNs and their derivatives dominated for years: Faster R-CNN introduced learnable region proposal networks [85], One-stage detectors such as SSD and YOLO emphasized real-time performance [86, 87]. CNNs remain highly effective for classification, detection, segmentation, keypoint estimation, and numerous domain-specific applications—including medical imaging and remote sensing—where their inductive biases toward locality and translation equivariance, combined with mature training methodologies, deliver competitive accuracy and strong computational efficiency.

1.4.2 Vision Transformers

Vision Transformers (ViTs) adapt Transformer encoders to images by processing sequences of patch tokens rather than raw pixels [75]. An image $x \in \mathbb{R}^{H\times W\times C}$ is divided into $P \times P$ patches, forming $N = \frac{HW}{P^2}$ vectors, which are linearly projected to D-dimensional embeddings:

$$\mathbf{z}_0 = \left[\mathbf{x}_{\text{class}}; \mathbf{x}_p^1\mathbf{E}; \mathbf{x}_p^2\mathbf{E}; \ldots; \mathbf{x}_p^N\mathbf{E}\right] + \mathbf{E}_{\text{pos}}, \tag{1.29}$$

where $\mathbf{x}_p^i \in \mathbb{R}^{P^2 \cdot C}$ is the i-th flattened image patch of size (P, P) with C channels, $\mathbf{E} \in \mathbb{R}^{(P^2 \cdot C) \times D}$ is the learnable projection matrix mapping patches to the D-dimensional embedding space, $\mathbf{x}_{\text{class}} \in \mathbb{R}^D$ is the learnable classification token (analogous to the [CLS] token in NLP), and $\mathbf{E}_{\text{pos}} \in \mathbb{R}^{(N+1) \times D}$ contains the positional embeddings for all N patches and the class token.

The resulting sequence $\mathbf{z}_0$ serves as the input to the Transformer encoder, where each token can exchange information with every other token via self-attention. This enables the model to integrate local patch features into a global image representation.

ViTs have scaled effectively to achieve state-of-the-art results across vision tasks. Architectural extensions have broadened the applicability of ViTs. DETR reframed object detection as direct set prediction with a Transformer backbone [88], while Swin Transformer introduced hierarchical features and windowed self-attention to scale efficiently to high-resolution images [89].

ViTs are now widely deployed in classification, detection, segmentation, and scene understanding, where their ability to capture long-range dependencies complements or surpasses the locality-focused inductive biases of CNNs.

CNNs encode strong inductive biases: locality via small kernels, translation equivariance via weight sharing, and a natural pyramidal hierarchy. These priors improve sample efficiency and stabilize training, which is advantageous in small to medium data regimes and for real-time deployment [84, 90]. ViTs rely on weaker priors and learn relations via attention, which provides direct global context and flexible feature interactions. This flexibility scales well with model size and data, often yielding superior performance at large scale [75, 91].

Empirically, ViTs tend to outperform CNNs when training data and compute budgets are sufficient, when global relationships matter, or when tasks benefit from long-range feature coupling, while CNNs remain competitive for compact models, low latency, and data-limited settings.

1.4.3 Single Image Super-Resolution (SISR)

Single image super-resolution (SISR) aims to reconstruct a plausible high-resolution (HR) image from a single low-resolution (LR) observation, an inherently ill-posed problem due to the many possible HR images that can correspond to a given LR measurement. Early deep-learning SISR methods such as SRCNN [92] and VDSR [93] followed a pre-upsampling design in which the LR image is first enlarged using a fixed interpolation method, typically bicubic, and then processed by a convolutional neural network (CNN) operating entirely in the HR domain. While effective, these approaches inherit the limitations of the initial interpolation step and incur high computational cost by applying convolutions to large HR tensors. This increased cost becomes particularly restrictive in scientific imaging scenarios where thousands of frames must be processed efficiently. Both SRCNN and VDSR are typically trained with a pixel-wise mean squared error (MSE) loss between the reconstructed SR im-

age I_{SR} and the ground-truth HR image I_{HR},

$$\mathscr{L}_{\mathrm{MSE}} = \frac{1}{N}\sum_{i=1}^{N}\left\|I_{\mathrm{HR}}(i) - I_{\mathrm{SR}}(i)\right\|_2^2, \tag{1.30}$$

which directly corresponds to maximizing the peak signal-to-noise ratio (PSNR),

$$\mathrm{PSNR} = 10\log_{10}\left(\frac{L_{\max}^2}{\mathscr{L}_{\mathrm{MSE}}}\right), \tag{1.31}$$

with $L_{\max}$ denoting the maximum possible pixel value (for example $L_{\max} = 255$ for 8-bit images).

Subsequent advances introduced generative adversarial network (GAN)–based models such as SRGAN [94] and ESRGAN [95], which optimize not only pixel-wise accuracy but also perceptual fidelity. These models incorporate adversarial losses and deep feature–based perceptual losses to produce visually convincing details that enhance the natural image realism of the super-resolved output. In SRGAN, for example, the generator loss typically combines a perceptual content term, defined as an MSE between VGG feature maps of I_{SR} and I_{HR}, with an adversarial term that encourages the discriminator to classify the generated image as real. This objective favors images that appear natural to a human observer rather than images that strictly preserve pixel intensities. However, such perceptual emphasis is not suitable for scientific measurement. GAN-based SR methods often hallucinate high-frequency textures that do not correspond to any real physical structure, and although these hallucinations improve perceptual realism, they degrade the interpretability and reliability of the reconstructed image for quantitative analysis. Furthermore, GAN architectures are computationally heavy, making them impractical when super-resolution must be applied to extensive high-frame-rate video datasets.

An alternative architecture that addresses the computational drawbacks of early CNN-based models while avoiding the perceptual risks of GAN-based methods is the Efficient Sub-Pixel Convolutional Network (ESPCN) proposed by Shi et al. [96]. ESPCN processes all feature extraction in the LR space and performs upsampling only at the final layer using a sub-pixel convolution (pixel-shuffle) operation. Like SRCNN, ESPCN is trained with a pixel-wise MSE loss between I_{SR} and I_{HR}, and therefore optimizes the same PSNR-oriented distortion measure, in contrast to SRGAN-style adversarial and perceptual losses. This purely pixel-level objective avoids hallucinated textures and preserves absolute intensity values, which is more appropriate for physically meaningful measurements. The ESPCN design eliminates the need for interpolation at the input stage, reduces computational and memory demands, and enables the network to learn an adaptive, data-driven upsampling filter. ESPCN achieves real-time performance for high-resolution video and produces sharper, more physically consistent edges, properties particularly important in domains where measurements depend on boundary accuracy rather than on perceptual texture.

In many experimental imaging contexts, including high-speed droplet dynamics, true HR ground-truth images are unavailable due to hardware limitations or the impracticality of repeating measurements under different optical configurations. This constraint has led to the development of self-supervised or zero-shot super-resolution frameworks such as ZSSR [97], which construct synthetic training pairs

internally by further downsampling the available LR image. These internal-learning methods allow super-resolution without access to HR ground truth but typically require training a separate model for each image, which is computationally infeasible for large video datasets and unsuitable when temporal consistency is essential.

Droplet imaging presents precisely this combination of constraints: the absence of HR ground truth, the need for computational efficiency across tens or hundreds of thousands of frames, and the requirement to preserve physically meaningful droplet boundaries without introducing artificial textures. All frames in such experiments originate from the same optical system and exhibit highly consistent image statistics, making dataset-level training more appropriate than per-image internal learning. For these reasons, efficient LR-domain architectures such as ESPCN, combined with self-supervised LR $\downarrow$ $\rightarrow$LR training strategies and pixel-wise distortion losses, offer a scientifically sound and computationally practical pathway for enhancing droplet video resolution while maintaining the physical fidelity necessary for accurate measurement of droplet width, curvature, and contact-line dynamics.

1.5 Sliding drops as a spatio-temporal problem

Many physical phenomena involve quantities that vary in both space and time. In the context of sliding drops, the drop shape, contact angles, and contact area can change as the drop moves over the substrate. Let $X_t \in \mathbb{R}^{H\times W\times C}$ denote the video frame at time t, with spatial resolution $H \times W$ and C channels. A generic spatio-temporal modeling problem can then be written as learning a mapping

$$f : \{X_1, X_2, \ldots, X_T\} \mapsto y \tag{1.32}$$

for scalar-on-sequence regression (for example, predicting a friction force or an effective material parameter), or

$$f : \{X_1, \ldots, X_T\} \mapsto \{\hat{Y}_1, \ldots, \hat{Y}_T\} \tag{1.33}$$

for sequence-to-sequence tasks such as forecasting future frames or contact-angle trajectories. This view connects directly to time series extrinsic regression, where the input is a full sequence but now each time step carries a spatial field rather than a scalar or low-dimensional vector.

Classical deep learning approaches to spatio-temporal data can be grouped into three broad families: three-dimensional convolutions that act jointly in space and time, architectures that separate spatial encoding and temporal modeling, and Transformer-based models that apply self-attention jointly over space and time.

A natural extension of two-dimensional convolution to video is the three-dimensional convolutional neural network (3D CNN). Here the convolution kernel has a temporal extent k_t in addition to spatial dimensions $k_h \times k_w$. A 3D convolutional layer maps an input tensor $x \in \mathbb{R}^{C\times T\times H\times W}$ to an output $y \in \mathbb{R}^{C'\times T'\times H'\times W'}$ via

$$y_{c',t,i,j} = b_{c'} + \sum_{c=1}^{C} \sum_{\tau=0}^{k_t-1} \sum_{u=0}^{k_h-1} \sum_{v=0}^{k_w-1} W_{c',c,\tau,u,v}\, x_{c,\, t+\tau,\, i+u,\, j+v}, \tag{1.34}$$

where $W_{c',c,\tau,u,v}$ are the learnable kernel weights and $b_{c'}$ is a bias term. By stacking such layers, 3D CNNs learn local spatio-temporal features such as moving edges,

deforming contours, or coherent motion patterns. Architectures like C3D demonstrated that 3D convolutions trained on large-scale video datasets can learn effective generic spatio-temporal features for action recognition and other video tasks [98].

An alternative strategy is to factor spatial and temporal processing. Two-stream convolutional networks process appearance and motion in parallel: one CNN operates on raw RGB frames to encode spatial cues, and a second CNN operates on stacks of optical-flow fields to encode short-term motion. The outputs of these streams are combined, for example by late fusion or a fully connected layer, to obtain an action or event prediction [99]. This separation has become a standard baseline in video analysis, illustrating that spatial and temporal information can be modeled by dedicated sub-networks that are fused at a later stage.

A more general form of spatial-temporal factorization is to use a convolutional network (or a vision Transformer) as a frame-level encoder and a recurrent model as a temporal aggregator. In long-term recurrent convolutional networks (LRCNs), a CNN extracts a feature vector z_t from each frame X_t, and this sequence $\{z_t\}_{t=1}^{T}$ is processed by an RNN or LSTM that produces sequence-level predictions [59]. This design instantiates the scalar-on-sequence regression viewpoint introduced above: the spatial encoder provides a compact representation of each frame, while the recurrent part captures temporal dependencies such as acceleration, oscillations, or history-dependent effects. Variants of this pattern are widely used for video classification, video captioning, and physical sequence modeling.

While standard LSTMs treat z_t as a one-dimensional feature vector, many physical systems are more naturally described as evolving spatial fields. Convolutional LSTMs (ConvLSTMs) address this by replacing the fully connected transformations in the LSTM gates with convolutions. Given an input sequence of feature maps $X_t \in \mathbb{R}^{C\times H\times W}$, the ConvLSTM update at time t reads

$$i_t = \sigma(W_{xi} * X_t + W_{hi} * H_{t-1} + b_i), \tag{1.35}$$
$$f_t = \sigma(W_{xf} * X_t + W_{hf} * H_{t-1} + b_f), \tag{1.36}$$
$$\tilde{C}_t = \tanh(W_{xc} * X_t + W_{hc} * H_{t-1} + b_c), \tag{1.37}$$
$$C_t = f_t \odot C_{t-1} + i_t \odot \tilde{C}_t, \tag{1.38}$$
$$o_t = \sigma(W_{xo} * X_t + W_{ho} * H_{t-1} + b_o), \tag{1.39}$$
$$H_t = o_t \odot \tanh(C_t), \tag{1.40}$$

where X_t and H_{t-1} are 3D tensors of size $C \times H \times W$, W_{xi} and W_{hi} (and analogously W_{xf}, W_{hf}, etc.) are learnable convolution kernels, and $*$ denotes convolution rather than matrix multiplication. Here H_t is the hidden state, C_t is the memory cell, and σ is the logistic sigmoid [100]. Because all gates operate on feature maps instead of flattened vectors, ConvLSTMs preserve spatial structure and can model how local patterns propagate and interact over time. This has been particularly successful in precipitation nowcasting and radar-based flow prediction, where both the input and the target are spatio-temporal fields [100].

More recently, Transformer architectures have been adapted to video. Instead of recurrent connections, these models rely on self-attention over space and time. TimeSformer extends the vision Transformer by dividing self-attention into separate spatial and temporal components, allowing each patch at location (x, y, t) to attend either to other patches in the same frame or to the same spatial location across

frames [101]. ViViT and related architectures treat a video as a sequence of spatio-temporal tokens and apply stacks of Transformer layers that learn global dependencies across both dimensions [102]. These models illustrate how self-attention can capture long-range temporal context and non-local spatial interactions without explicit recurrence.

For sliding-drop experiments, the choice of spatio-temporal model depends on data characteristics and the target quantity. When high-resolution videos are available and large training datasets can be collected, 3D CNNs or video Transformers can operate directly on frame sequences and learn rich volumetric features of the moving interface. When data are limited or the spatial resolution is low, a more compact approach is to extract per-frame features with a CNN (such as drop width, length, contact-line region descriptors, or intermediate feature maps) and apply an RNN, LSTM, or Transformer encoder along the temporal dimension. This reduces the dimensionality of the spatio-temporal input while still leveraging the temporal modeling tools described in the previous sections. In all cases, the goal of spatio-temporal modeling is to link the visual evolution of the drop to physically meaningful quantities, such as friction force, in a way that respects both the spatial structure of the interface and its temporal dynamics.

1.6 Methodological roadmap

The methodological review in this chapter is intended to survey common approaches in the literature and to establish a shared toolkit for the three studies that follow, clarifying how the thesis progresses from physical observables to learnable representations. The first paper combines image restoration and geometric post-processing to improve the extraction of dynamic contact angles from limited-resolution recordings, since the relevant contact-line features are often degraded by blur, noise, and sampling limits, and enforcing consistent geometric constraints improves the stability of the extracted contours and angles across time. The second paper builds on these extracted observables and treats sliding as a multivariate time-series problem, where recurrent sequence models are a natural choice for learning temporal dependencies and estimating quantities that are not directly accessible from a single view. The third paper then generalizes this idea to an end-to-end spatio-temporal formulation that operates directly on sequences of frames and integrates convolutional feature extraction with attention-based temporal modeling. To keep the discussion concise and focused, we emphasize methods that are used directly in this thesis or are most closely related, while many other viable alternatives (e.g., Hidden Markov Models, ARIMA-type time-series models, or polynomial fitting approaches) are not discussed in detail. In this way, the methods introduced in Sections 1.2–1.5 reappear throughout the dissertation as progressively stronger levels of integration, from physics-guided measurement to feature-based sequence learning and finally to fully data-driven spatio-temporal modeling.

Chapter 2

Publications Overview

2.1 Overview of the cumulative dissertation

This cumulative dissertation presents research that investigates how machine learning methods can be used to extract reliable quantitative information from high-speed videos of sliding drops. The overarching goal is to turn complex, noisy video data into precise geometric and physical quantities that are directly relevant for surface science, such as dynamic contact angles and the width of sliding drops along their full trajectory.

The research addresses two central challenges. First, sliding drops occupy only a small fraction of each video frame, and limited spatial resolution at the contact line makes accurate contact angle measurements difficult. Second, the friction force of a sliding drop depends on its width, yet direct front-view measurements require additional cameras or mirrors and restrict the observable sliding length. These limitations motivate data-driven methods that can infer width from side-view recordings alone, without changing the experimental setup.

The dissertation develops this research program in three main steps. The first work introduces a deep learning based super resolution and polynomial fitting framework that enables accurate extraction of advancing and receding contact angles from low-resolution side-view videos of sliding drops and packages the full workflow in the 4S-SROF toolkit. Building on this foundation, the second work formulates the estimation of the front-view drop width as a multivariate time-series regression problem that uses side-view features such as dynamic contact angles, drop geometry, and velocity, and systematically compares classical regression and recurrent neural network models. The third work removes the reliance on predefined features and proposes an end-to-end spatiotemporal CNN-Transformer architecture with position-invariant video processing and a low-dimensional absolute positional encoding that operates directly on sequences of side-view video frames.

Together, these publications construct a coherent methodological pipeline for machine learning-based analysis of sliding drops, starting from improved feature extraction, moving to learning-based width estimation from side-view measurements, and leading to an optimized CNN-Transformer tailored to low-dimensional time series derived from video data. The individual studies are presented in the appendix of this dissertation, while the present chapter summarizes their content and clarifies the author's contributions.

2.2 Included publications

This cumulative dissertation is based on the following peer-reviewed publications, which are referred to as P1–P3 in the remainder of this chapter.

2.2.1 P1: Deep Learning to Analyze Sliding Drops

P1 S. Shumaly, F. Darvish, X. Li, A. Saal, C. Hinduja, W. Steffen, O. Kukharenko, H. J. Butt, and R. Berger, "Deep Learning to Analyze Sliding Drops," *Langmuir*, 39, 1111–1122, 2023 [103].

In this work, a super-resolution approach based on an Efficient Sub-Pixel Convolutional Network (ESPCN) is combined with an optimized polynomial fitting strategy to improve the accuracy of dynamic contact angle measurements in videos of drops sliding down a tilted plate. The study demonstrates that upscaling low-resolution cutouts of the sliding drops by a factor of three in each spatial dimension leads to substantially sharper contours and enables more reliable extraction of advancing and receding contact angles. A systematic parameter study with synthetic reference data is used to select a suitable polynomial order and window size. The full workflow, including baseline detection, contour extraction, super-resolution, and contact angle fitting, is integrated into the 4S SROF toolkit, which also provides additional geometric quantities such as drop length, median line angle, and velocity.

2.2.2 P2: Estimating sliding drop width via side view features using recurrent neural networks

P2 S. Shumaly, F. Darvish, X. Li, O. Kukharenko, W. Steffen, Y. Guo, H. J. Butt, and R. Berger, "Estimating sliding drop width via side view features using recurrent neural networks," *Scientific Reports*, 14, 12033, 2024 [104].

The second publication addresses the problem of estimating the width of sliding drops, which is a key parameter for friction force calculation. Using the 4S-SROF toolkit developed in P1, the authors extract time series of physically meaningful features such as dynamic advancing and receding contact angles, drop length, height, middle line angle, and velocity. These multivariate sequences are used to train and compare a range of regression and multivariate sequence analysis models, including linear regression, random forests, convolutional networks, recurrent neural networks, LSTM, GRU, bidirectional LSTM, and ConvLSTM. The study shows that an LSTM model with a sliding window of 20 frames yields the best performance, achieving a root mean square error of approximately 67 μm, corresponding to a relative error of about 2.4 percent over a width range between 1.6 and 4.4 mm. The approach allows the width to be estimated continuously along a sliding length of about 5 cm, without additional cameras or mirrors.

2.2.3 P3: CNN–Transformer with Absolute Positional Encoding Optimized for Low Dimensional Inputs

P3 S. Shumaly, F. Darvish, M. Salehi, N. M. Foumani, O. Kukharenko, H. J. Butt, U. Schwanecke, and R. Berger, "CNN–Transformer with Absolute Positional Encoding Opti-

mized for Low Dimensional Inputs: Applied to Estimate Sliding Drop Width," in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases (ECML–PKDD)*, Lecture Notes in Artificial Intelligence 16021, Springer Nature Switzerland, 2026 [105]. The work was accepted for presentation at ECML–PKDD 2025.

The third publication introduces an end-to-end deep learning framework that operates directly on sequences of side-view video frames and estimates the millimetre-scale width of sliding drops without predefined features. The method first applies a position invariant video processing scheme based on a sliding spatiotemporal window that keeps the drop centred and discards irrelevant background, thereby reducing computation time and memory consumption by more than eighty percent while preserving the region of interest. A VGG8-inspired convolutional backbone extracts compact spatial representations, which are then passed to a time series transformer (ConvTran). To make attention-based processing effective in the low-dimensional regime. The study proposes a low-dimensional absolute positional encoding (ldAPE) that improves the dot product of the encoding for small feature spaces and empirically outperforms standard positional encodings on 32-dimensional inputs. The resulting CNN-Transformer model achieves a root mean square error of about 48 μm, corresponding to a relative error of roughly 1.7 percent, and provides Grad-CAM-based interpretability of the regions that are most informative for the prediction. Code and data are released as an open-source toolkit.

2.3 Relation between the publications

The three publications form a coherent research trajectory that gradually increases the level of automation and representation learning in the analysis of sliding drop experiments.

Publication P1 develops the experimental and algorithmic foundation required for any subsequent machine learning on sliding drop videos. It tackles the fundamental resolution limitation by training a super-resolution network specifically on drop images and by systematically optimising the polynomial fitting procedure for contact angle extraction. The outcome is a robust pipeline that can transform raw high-speed side-view videos into accurate time series of geometric quantities such as advancing and receding contact angles, drop length, and velocity. This toolkit is used both in later works and in the wider surface science community.

Publication P2 takes these derived time series as input and formulates the estimation of the front view drop width as a time series extrinsic regression problem. Instead of relying on empirical correlations involving a single scalar such as the centre velocity, the study explores multivariate dependencies between several side-view features and the unknown width and compares a broad range of regression and sequence models. The systematic evaluation clarifies the benefits of recurrent architectures for this type of data. P2 provides a first machine learning based solution to continuous width estimation along the full sliding path without additional experimental hardware.

Publication P3 builds on the insights and limitations identified in P2. While P2 depends on hand-engineered features extracted by a separate tool, P3 removes this separation and lets the model operate directly on raw video data. The position invariant spatiotemporal window mechanism addresses the problem that sliding

drops occupy only a small part of the frame and that naive cropping can introduce positional bias. The combination of a compact VGG8-style backbone and ConvTran allows the architecture to capture both local contour details and long-range temporal dependencies in an efficient way. The newly proposed low-dimensional absolute positional encoding addresses a specific bottleneck of transformer models applied to small feature spaces and provides both theoretical and empirical improvements for this regime. As a result, P3 achieves higher accuracy than P2 and offers improved interpretability through attention and Grad CAM analysis.

Viewed together, the publications progress from improving measurement quality (P1), through learning from carefully extracted physical features (P2), to a fully end-to-end video-based CNN–Transformer that incorporates positional information and spatiotemporal context directly (P3). They demonstrate that machine learning can not only replace additional cameras and optical components, but can also reveal which parts of the drop contour and which temporal segments are most informative for understanding friction and wetting behaviour.

Chapter 3

Publication 1

3.1 Deep Learning to Analyze Sliding Drops

3.1.1 Summary and author contribution

This publication establishes the experimental and computational basis for the subsequent machine-learning analyses of sliding-drop videos. It addresses the key limitation of insufficient spatial resolution by training a super-resolution model tailored to droplet imagery and systematically refining the polynomial fitting strategy used for contact-angle determination. Together, these steps yield a robust processing pipeline that converts raw high-speed side-view recordings into reliable time series of geometric observables, including advancing and receding contact angles, drop length, and velocity. The resulting toolkit underpins the later publications in this dissertation and supports broader use in surface-science workflows.

The author contributed to the conception and design of the study, with a focus on applying deep-learning methods to the quantitative analysis of sliding-drop videos. The author developed and implemented the computational pipeline for data preprocessing, model training, and evaluation, and participated in designing the experiments and analyzing the results. Major portions of the manuscript related to the machine-learning methodology and interpretation of outcomes were written and revised by the author.

3.1.2 Scientific publication

LANGMUIR

pubs.acs.org/Langmuir Article

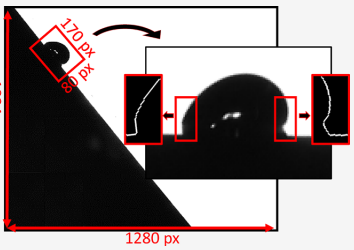
Deep Learning to Analyze Sliding Drops

Sajjad Shumaly, Fahimeh Darvish, Xiaomei Li, Alexander Saal, Chirag Hinduja, Werner Steffen, Oleksandra Kukharenko, Hans-Jürgen Butt, and Rüdiger Berger*

Cite This: *Langmuir* 2023, 39, 1111−1122 Read Online

ACCESS | Metrics & More | Article Recommendations | Supporting Information

ABSTRACT: State-of-the-art contact angle measurements usually involve image analysis of sessile drops. The drops are symmetric and images can be taken at high resolution. The analysis of videos of drops sliding down a tilted plate is hampered due to the low resolution of the cutout area where the drop is visible. The challenge is to analyze all video images automatically, while the drops are not symmetric anymore and contact angles change while sliding down the tilted plate. To increase the accuracy of contact angles, we present a 4-segment super-resolution optimized-fitting (4S-SROF) method. We developed a deep learning-based super-resolution model with an upscale ratio of 3; i.e., the trained model is able to enlarge drop images 9 times accurately (PSNR = 36.39). In addition, a systematic experiment using synthetic images was conducted to determine the best parameters for polynomial fitting of contact angles. Our method improved the accuracy by 21% for contact angles lower than 90° and by 33% for contact angles higher than 90°.



INTRODUCTION

Sessile drops on solid surfaces assume a semispherical shape to attain a state of minimal energy.[1−4] The shape of sessile drops is axisymmetric, and it can be fitted with a solution of the Laplace equation.[5−7] In a real wetting situation, the contact line is trapped in a metastable state. The contact angle (CA) lies in a range between the advancing CA (θ_a) and the receding CA (θ_r), depending on how the drop is placed on the surface.[8] Therefore, the θ_a and θ_r are used as representative parameters that describe substrate wettability and surface tension.[9,10] The θ_a represents the angle at which the liquid advances over a solid surface and the θ_r is the angle of a receding contact line.[11] The CAs extracted are apparent CAs between the solid−liquid and the liquid−air interfaces at the contact line. The difference between both angles is called CA hysteresis.[12]

When drops move, for example, down a tilted plate, they are not axisymmetric anymore. They become more and more elongated depending on velocity. In addition, the CAs become dynamic CAs. For simplicity, we keep the terms advancing and receding CAs. At the front side, they assume the θ_a, and at the rear side, the θ_r.

A single drop takes up just a tiny part of each video frame since the entire slide path of a drop must be recorded. Focusing with a higher magnification objective on the droplet in a specific position would result in a loss of information for the trajectory of the sliding drop. As a result, the resolution at which the drop contour and three-phase contact line can be resolved is limited by the pixel size. In a typical example, the captured image has a dimension of 1280 by 1024 pixels, while the average dimension of the drop is only 170 by 80 pixels (Figure 1A). Thus, the first and most important challenge in analyzing sliding drops on a tilted plate is to enhance resolution (Figure 1B−D). Extracting CAs can be more accurate using super-resolution images (Figure 1E).

For a sliding drop, the θ_a usually increases, while the θ_r decreases with the increase of velocity. Therefore, a method that calculates θ_a and θ_r separately is required in order to measure dynamic CAs. Based on the Weierstrass approximation theorem, polynomials can be used to approximate uniformly any continuous function of a single variable defined on a closed interval.[13] Researchers have demonstrated that accurate CAs can be obtained for symmetric and asymmetric situations using polynomial fitting.[14,15] However, a wide range of CAs cannot be calculated with a specific polynomial order or a predetermined number of pixels of the drop surface.[16] It means that the order of the polynomial and the number of input pixels are two crucial parameters to obtain accurate CAs. So far, the best results have been reported for polynomial order ranging between two and four.[14,15,17] Most reports, however, lack a proper analysis of standards or references to estimate the accuracy. Correlation coefficients and standard deviations of the results were given to evaluate the quality of fitting. Those errors based on reproducibility without existing standard reference may lead to systematic errors in measuring CAs. For asymmetric drops, the error can be up to 5° using polynomial fitting approaches[15] and they are sensitive to drop image resolution. Methods based on polynomials can measure the

Received: October 18, 2022
Revised: December 19, 2022
Published: January 12, 2023

https://doi.org/10.1021/acs.langmuir.2c02847
Langmuir 2023, 39, 1111−1122

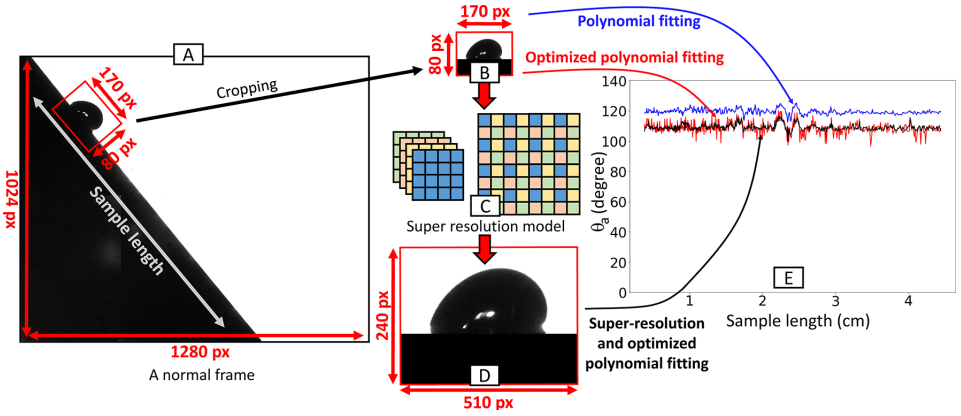


Figure 1. Example snapshot from a recorded video and processing routine presented in this article. (A) Full image resolution (1280 by 1024 pixels). (B) Original resolution of the extracted drop image (170 by 80 pixels). (C) Illustration of the applied super-resolution model. (D) Drop image with increased resolution (240 by 510 pixels). (E) Comparison of the calculated advancing angle on the original image with polynomial fitting (blue), with the optimized polynomial fitting presented in this paper (red), on the drop with the optimized fitting on the super-resolution image (black).

CAs of both sides of the drop independently. In practice, however, the implementation is complicated because parameters need to be extracted based on the problem conditions. Here, we suggest a method for quantitative estimation of fitting errors based on the analytical expressions of artificially generated reference drops. This approach will be used to extract polynomial parameters based on the optimization of the fitting errors. The accuracy of fitting methods is highly dependent on the image resolution.

Recovering a high-resolution image or video from its low-resolution counterpart is an active area in digital image processing.[18] It is referred to as super-resolution. Super-resolution can be divided into single-image super-resolution (SISR) or multi-image super-resolution (MISR). Increasing sliding drop resolution is a SISR problem. The SISR problem is ill-posed because one low-resolution image may correspond to multiple high-resolution image solutions. There are three main categories of SISR algorithms: interpolation-based, reconstruction-based, and learning-based.[19] Interpolation-based SISR methods are fast but not very accurate. Bicubic interpolation is the most popular method in this category.[20] Reconstruction-based methods are the second category in which sophisticated prior knowledge is the basis for these methods.[21,22] They are slow and their accuracy is very sensitive to the scale factor. Learning-based SISR methods are attracting attention due to their high speed and accuracy. These methods are based on machine learning algorithms and learn from training samples to interpret relationships between low-resolution and high-resolution images. A branch of machine learning algorithms called deep learning[23] is able to learn informative hierarchical representations automatically. Deep learning algorithms perform better than traditional machine learning algorithms across numerous fields: deep learning algorithms in the SISR are applied, for example, in medical imaging,[24] fluorescence microscopy in biology,[25,26] atomic force microscopy,[27] FIB-SEM[28] in materials science, and reconstruction of turbulent flows in physics.[29,30]

Convolutional neural network (CNN) is one of the most successful subsets of deep learning. It is primarily used to process images.[23] Based on CNN, many super-resolution models have been proposed.[31−34] In all mentioned models, before reconstruction, the input image was upscaled to a high-resolution space by using a single filter, typically bicubic interpolation. In other words, the super-resolution operation takes place in the high-resolution space, which increases computational complexity substantially. To solve this problem, Shi et al.[35] introduced the Efficient Sub-Pixel Convolutional Neural Network (ESPCN). In ESPCN, there is an efficient sub-pixel convolution layer that learns an array of upscaling filters to upscale the final low-resolution feature maps and turn them into the high-resolution output. By replacing the handcrafted bicubic filter with a trainable layer and moving this layer to the very end of the network, they succeeded in increasing accuracy and decreasing computational complexity. Due to its efficiency and speed, ESPCN is well suited for real-time image processing and especially video analysis.

Here, we present a method to enhance the accuracy in CA determination by enhancing the resolution of video frames and by optimizing a polynomial fitting. After recording drops moving by a high-speed camera, we processed each frame of the video to extract the drop profile. Here, we faced a number of challenges: How can we increase the accuracy of CA measurements from low-resolution videos? How can we measure the CA for asymmetric and deformed drops as accurately as possible? How can we determine the improvement in the accuracy of the measured CAs (since there is no reference method for drops on a tilted plate)?

We trained an ESPCN super-resolution model with an upscale ratio of 3; i.e., the trained model was able to enlarge drop images 9 times with high accuracy. Then, we optimized a flexible polynomial fitting to measure dynamic advancing and receding CAs separately. To examine the accuracy, we conducted a systematic experiment using synthetic images.

We propose a toolkit to extract drop profiles from a high-speed camera based on a modified ESPCN super-resolution model and optimized polynomial fitting. This toolkit gains all relevant parameters such as advancing CA, receding CA, drop length, median line angle, and velocity from the videos.

https://doi.org/10.1021/acs.langmuir.2c02847
Langmuir 2023, 39, 1111−1122

MATERIALS AND METHODS

Tilted Plate Experiments. Deionized water drops were placed on top of a tilted plate using a peristaltic pump connected to a grounded syringe needle. A high-speed camera (FASTCAM Mini UX100 (Photron) with a TitanTL telecentric lens, ×0.268, one inch, C-mount (Edmund Optics)) captured videos of sliding drops from the side view. The illumination conditions were controlled by a telecentric backlight illuminator (138 mm, Edmund Optics). Typically, the imaged slide length corresponds to 4.5 cm in all measurements. The experimental temperatures were 20 ± 1 °C and humidity levels were 15−30%.

Sample Preparation. In this study, we performed some sliding drop experiments on hydrophobic samples with a point defect and samples with a chemical heterogeneity (a strip perpendicular to the sliding direction). For a better understanding, the schematics of both samples are represented in Figure S1a.

Samples with a Topographic Defect. As substrates, 170 μm thick precision glass coverslips were used (Carl Roth no. 1.5H).[36] Water, ethanol, and acetone were used to clean the coverslips. To prepare topographic defects, we used photolithography (masks provided by DeltaMask). The defect was a SU8 cylindrical pillar with a diameter of 1100 μm and a height of 10 μm. After cleaning the substrates in isopropanol, O_2-plasma (Diener Electronic, Femto BLS) was used to activate the substrates for 1 min at 30 W, a flow rate of 0.3 cm^3/s, and a pressure of 0.3 mbar. Next, the surface was exposed to the vapor of trichloro(1*H*,1*H*,2*H*,2*H*-perfluorooctyl)silane (PFOTS, SIGMA-ALDRICH Chemie GmbH, 97%). In detail, 100 μL of PFOTS liquid was placed in a desiccator (cylindrical volume, the diameter was 20 cm, and the depth was 15 cm) at 50 mbar with the samples. Samples were placed 5 cm above the PFOTS liquid container. Then, the container was placed on a magnetic stirring plate. At the end, the samples were removed from the container after 30 min.

Samples with a Chemical Heterogeneity. Chemically heterogeneous samples have been prepared by the process of double chemical vapor deposition with the use of a glass mask.[37] Standard microscopic glass slides were cleaned with Milli-Q water, then with acetone, ethanol, and 5 min oxygen plasma treatment at 300 W and 0.3 bar (Diener Electronic Femto). Then, they were treated by chemical vapor deposition of PFOTS in a desiccator. PFOTS liquid (1 mL) was placed in the desiccator at less than 20 mbar for 10 min. The pump was switched off and the samples remained for 20 min. They were transferred to a vacuum oven tempered at 25 °C for another 10 min. A glass shadow mask was placed over them in an O_2-plasma chamber (300 W, 0.3 bar) for 5 min. The uncovered portion of the PFOTS layer was obliterated with this exposure. Chemical vapor deposition began immediately in the desiccator containing 200 μL of octyltrichlorosilane (OTS, SIGMA-ALDRICH Chemie GmbH, 97%). At the end, the samples remained in the vacuum for 120 min at 150 mbar. Due to the complexity of the preparation process, a schematic of all steps is represented in Figure S1b.

Super-Resolution Model. To increase the accuracy of θ_a and θ_r, we increased image resolution using the ESPCN super-resolution model. Our input files were videos, which contained at least 200 frames of a drop in different positions. To study drop charges, researchers may need to analyze videos containing 100 subsequent drops.[38] Thus, the calculation speed is a crucial factor for analysis of hundreds of videos in a row. We selected ESPCN because of its high processing speed and high accuracy.[35] The dataset, ESPCN architecture and parameters, and training procedure are described below.

For training the super-resolution model, 1400 videos of sliding drops were gathered. Using a high-speed camera, subsequent frames in a video exhibit nearly identical drop shapes. It is however desirable to have different shapes of drops for training the super-resolution model. Thus, 10 frames were extracted from each video. The final dataset consisted of 14,000 images from sliding drops imaged under different conditions (Figure S2; the dataset is downloadable on GitHub[39]). The Keras[40] and TensorFlow[41] libraries were used to train the ESPCN model. We considered training, validation, and test sets as 80, 10, and 10%. Test set contains 1400 frames from different videos, completely separate from training and validation sets.

Before training, we needed to define the scale factor chosen to increase resolution. A scale factor of x indicates that the trained model enlarges drop images x times for each axis. The whole image is accordingly scaled by x^2. In order to find out which scale factor is best, we trained three models by scale factors 2, 3, and 4. We used peak-signal-to-noise ratio (PSNR, units: dB) as a parameter to evaluate the quality of the reconstructed images.[33] As the scale factor increases, the image resolution increases. However, the accuracy (PSNR) decreases (Figure S3). Therefore, we repeated all results presented in Section 3.2 for scale factors 2, 3, and 4. As a result, a scale factor of 3 is the best choice, especially for CAs higher than 90°.

The architecture of the ESPCN model has two layers with a depth of 64 and 32 nodes as hidden layers for feature mapping operations. The dimensions of the input images are not constant. We had widths ranging from 129 to 264 px and heights ranging from 49 to 102 px. There is another layer called the sub-pixel convolution layer to construct the super-resolution image as the output. The activation function for the ESPCN model was tanh. Reconstructed images are more precise when the PSNR is high. The PSNR was calculated by first calculating the one-dimensional mean squared error:

$$\mathrm{MSE}_{1D} = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2 \tag{1}$$

Here, n is the number of data points, Y_i are the observed values, and $\hat{Y}_i$ are the predicted values. To compare two matrices/images, we define the MSE_{2D} in a similar way by

$$\mathrm{MSE}_{2D} = \frac{1}{m\,n}\sum_{i=1}^{m}\sum_{j=1}^{n}(Y_{(i,j)} - \hat{Y}_{(i,j)})^2 \tag{2}$$

$$\mathrm{PSNR} = 10\log_{10}\left(\frac{\mathrm{MAX}_I^2}{\mathrm{MSE}}\right) \tag{3}$$

Here, MAX_I is the maximum possible pixel value of the reference image, $Y_{(i,j)}$ are the pixel values of the reference image, and $\hat{Y}_{(i,j)}$ are the pixel values of the predicted image.

In the training process, the reference images are based on the scale factor (here is 3). Then, the super-resolution model tries to return the downsized image to the reference resolution. In this way, the model is able to compare the predicted image and the reference image and start to learn. $Y_{(i,j)}$ and $\hat{Y}_{(i,j)}$ are calculated after the training process on the test data. The model accuracy based on PSNR on 400 epochs for a scale factor of 3 was calculated at 35.46. An epoch is one cycle of training with all the training data. We have chosen 400 epochs because the training process diagram plateaued after 300 epochs (Figure S4). We modified the ESPCN architecture and added a layer with a depth of 64 nodes between the hidden layers and changed the activation function to ReLU (Figure S5). As a result of the modifications, the PSNR improved from 35.46 to 36.39 for the same scale factor and number of epochs. Since the PSNR is a logarithmic measure,[42] differences in the order of 1 are noteworthy. In the training process of the modified ESPCN model, the training and the validation curves were compatible (Figure S4). It means that the training process has been done correctly. A common traditional model called bicubic was also used to increase the resolution and calculate related PSNR to compare it to the proposed model. The PSNR for the bicubic model was 28.90. The difference between the modified ESPCN and the bicubic model is considerable. In all cases, the modified ESPCN obtained a considerably better PSNR than bicubic (Figure S6).

We compared a representative image from a video file before and after applying the ESPCN algorithm visually and after applying the bicubic method (Figure 2). In all cases, we applied a canny edge detection algorithm[43] implemented by OpenCV.[44] The drop contour after application of ESPCN appears sharper with more details (Figure S7). The reason is that canny edge detection operates in a sub-pixel environment after using the super-resolution model.[35] Increasing

https://doi.org/10.1021/acs.langmuir.2c02847
Langmuir 2023, 39, 1111−1122

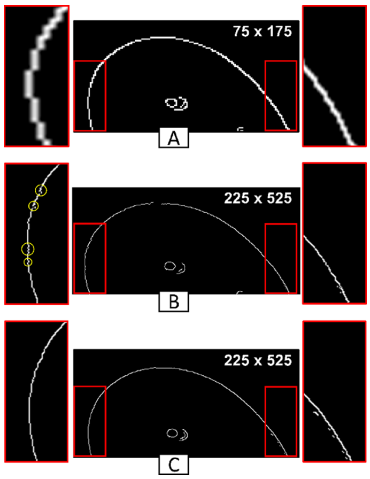


Figure 2. Examples of accuracies of edge detection on a drop image before using super-resolution (A), after using the bicubic method (B), and after using the super-resolution model (C). (A) The drop image in the original resolution (75 × 175 pixels). (B) The upscaled image using the bicubic method (225 × 525 pixels). The drop curve is smoother than the low-resolution image, but some edge detection displacements are visible, especially for the advancing part (yellow circles). The edge for the upper part of the drop is not detected properly, and the drop contour is not connected. (C) After applying the super-resolution model, the image has 225 × 525 pixels. Visually, the extracted edge of a super-resolution image has a very smooth curve. All mentioned drop contours were created after detecting the baseline and cropping the upside part.

image details by using sub-pixel algorithms is a common approach to measuring CAs.[16,45,46] Prior to using the super-resolution model, baseline detection is performed (details shown in the Supporting Information). Thus, the drop image was extracted without reflection before being fed into the super-resolution model.

Although the ESPCN image appears sharper, it is not clear if it will lead to a more precise extraction of the CA. To answer the question, CA improvement is discussed in the following sections.

Polynomial Fitting. CAs can be extracted by fitting an ellipse to the drop contour when the drop is sliding smoothly.[47] In general, the shape of a sliding drop is non-elliptic, and it is non-symmetric with respect to the front and rear. Moreover, the drop's shape may be deformed at the advancing and receding side due to interaction with defects.[48] Therefore, we aim to develop a universal method to calculate CAs. The polynomial fitting can be used to separately calculate θ_a and θ_r based on adjacent pixels.

The accuracy of polynomial fitting depends on the order of the polynomial and the number of pixels taken from a contour line as input. As an example, we show how the determination of θ_a depends on the pixel number and the polynomial order (Figure 3). The first column shows the position of the drop contour based on the edge detection algorithm without polynomial fitting. The second column represents a fit of the drop contour with a line (polynomial order one), the third column a fit with a third-order polynomial, and the fourth column with an order of eight. For each fit, we varied the length of the contour line, i.e., 20 pixels (1st row), 40 pixels (2nd row), and 70 pixels (3rd row).

A CA with less than 20 pixels input can be approximated with a linear fit for extracting CAs, while more pixels overestimate the CA in this example. For the eighth-order polynomial, the fit follows all pixels without following a general drop contour. An appropriate polynomial order in our case can be close to the third-order polynomial, which generally follows the drop curve for all ranges of input pixels. Still, a

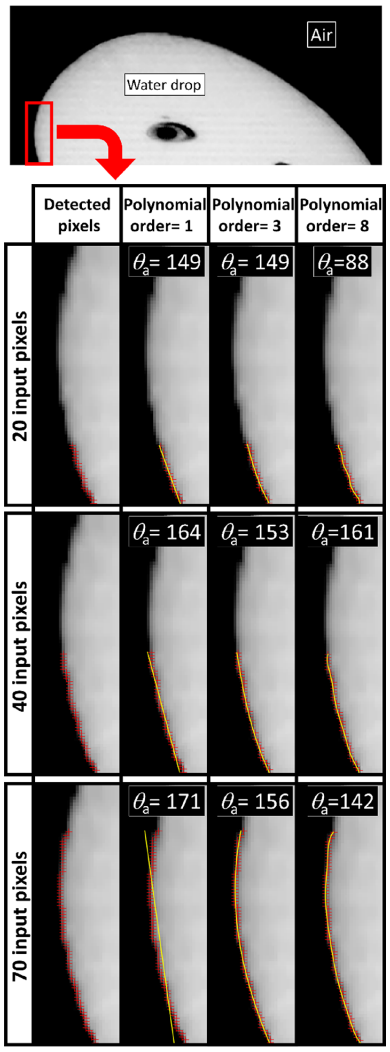


Figure 3. Differences in the calculated advancing angle (θ_a) after polynomial fitting with varying parameters: numbers of pixels (20, 40, 70) and orders of the polynomial (1, 3, 8). The red plus symbols represent the selected pixels. The yellow curve represents the fitted polynomial. There is a black area in the center of the drop image that is just a reflection that appears in all recorded frames. When p = 1, a line is fitted to the pixels. Thus, when the number of input pixels increases, the fitting cannot follow the pixels. When p = 8, the polynomial follows all pixels and there is no generalization. For p = 3, the polynomial follows the drop shape. In this case, measurement variations for different input pixels are lower than other mentioned polynomials.

variation in a CA of 7° results from an increase in input pixels from 20 to 70. An appropriate selection of input pixels and polynomial order can even help to reduce optical noise (Figure S8). Choosing the polynomial order that is most accurate may differ based on the circumstances of each problem. The question arises, which order of the polynomial fit function (p) and the number of input pixels (n) are required to extract the correct CA value? In addition, we expect differences for n and p for CAs <90° and CAs >90°.

https://doi.org/10.1021/acs.langmuir.2c02847
Langmuir 2023, 39, 1111−1122

RESULTS AND DISCUSSION

Does the ESPCN model increase the accuracy of the extracted CA? How many pixels and what polynomial order is the best for sliding drops on different surfaces? To answer these questions, a defined reference is required to calculate the accuracy of the polynomial results by changing the above variables. We use synthetic images[14] but not containing reflections to calculate the accuracy.

Reference Construction. A dashed box represents the screen and a circle represents a drop (Figure 4). The contour of a drop is simulated by the part of the circle that is inside the dashed box. It is possible to generate all ranges of the CAs by changing the radius of the circle and the size of the box. The images were generated in Python using the OpenCV image processing library.[44] The *y*-axis direction in image processing is downward. Accordingly, the origin of the produced images is upside left, and the derived formulas are considered in the fourth quadrant of Cartesian coordinates.

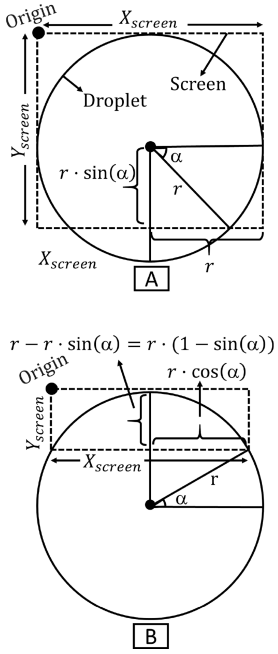


Figure 4. A visual representation of the components of the proposed approach for creating synthetic images. (A) To produce synthetic images for CAs >90°. (B) To produce synthetic images for CAs <90°.

In our case, experimental images of sliding drops extracted from videos have an average of 13,600 pixels.

$$X_{screen} \cdot Y_{screen} = 13600 \tag{4}$$

To match experiments, the number of pixels should be identical to the synthetic image. Therefore, eq 4 expresses the product of the vertical and horizontal number of pixels of the synthetic images.

$$X_{screen} = 2r_{PO} \tag{5}$$

$$Y_{screen} = r_{PO} \cdot (1 + \sin(\alpha_{PO})) \tag{6}$$

$$r_{PO} = \sqrt{\frac{13600}{2(1 + \sin(\alpha_{PO}))}} \tag{7}$$

Equation 7 is obtained by substituting eqs 5 and 6 into eq 4. Also, $(X - r_{PO})^2 + (Y + r_{PO})^2 = {r_{PO}}^2$ represents an equation of a circle for the hydrophobic part. Using this equation and eq 7, it is possible to generate drops with a CA higher than 90°.

$$X_{screen} = 2r_{PI} \cdot \cos(\alpha_{PI}) \tag{8}$$

$$Y_{screen} = r_{PI} \cdot (1 - \sin(\alpha_{PI})) \tag{9}$$

$$r_{PI} = \sqrt{\frac{13600}{2\cos(\alpha_{PI}) \cdot (1 - \sin(\alpha_{PI}))}} \tag{10}$$

Equation 10 is obtained by substituting eqs 8 and 9 into eq 4. Also, $(X - r_{PI} \cdot \cos(\alpha_{PI}))^2 + (Y + r_{PI})^2 = {r_{PI}}^2$ represents an equation of a circle for the hydrophilic part. Using this equation and eq 10, it is possible to generate drops with a CA lower than 90°.

PO and PI are abbreviations for hydrophobe and hydrophile. Here, r is the radius of the synthetic drop, and α is the angle between the horizontal line and the line that connects the center of the drop to the intersection of the circle and the box. The Y_{screen} is the vertical number of screen pixels, and the X_{screen} is the horizontal number of screen pixels. Following image generation, a Gaussian filter was applied to smooth the contour of the drop. Examples of synthetic images generated using the mentioned formulas when α was equal to 30° and 60° for CAs >90° and CAs <90° are provided in the SI (Figure S9).

We used synthetic images to simulate asymmetrical drop images by taking two halves of an image. One half represents the advancing side and the other half is the receding side of the drop. In general, $\theta_a > \theta_r$ (Figure S10).

Optimization of Polynomial Variables. MSEs were calculated for low-resolution artificial images and super-resolution images after the super-resolution model was applied. Conceptually, we distinguish hydrophilic (15−90°) and hydrophobic (90−165°) surfaces by CAs. Hereby, a calculated MSE corresponds to the related range of possible CAs for CAs <90° and CAs >90°. The MSEs (eq 1) for different polynomial orders from 1 to 10 were calculated. The number of input pixels for both low and super-resolutions starts from 5 px with a 5 px increment.

For CAs <90° and low-resolution images, the minimum MSE is 3.41. This value corresponds to the order of a polynomial $p = 1$ and the number of input pixels $n = 10$ (Figure 5A). Thus, for hydrophilic samples, a fit of a tangent close to the three-phase contact line is the best approach. Increasing the polynomial order increases the minimal error values to 9.80, when $p = 2$, and to 15.51 for $p = 3$ (Figure 5E, black line). For CAs >90° and low-resolution images, the minimum MSE is 10.9. This error corresponds to $p = 4$ and $n = 60$ (Figure 5B). Also, p as 3 and 5 are good options since their MSEs are close to the minimum (Figure 5F, black line). This analysis of low-resolution images indicates that the choice of the best combination of p and n is less obvious for CAs >90°.

For CAs <90° and super-resolution images, the lowest error was 2.82, which corresponds to $p = 2$ and $n = 65$ (Figure 5C). Thus, the error of the super-resolution image is lower than the error of the tangent fitting ($p = 1$) in the low-resolution images. Taking the super-resolution images for hydrophobic

https://doi.org/10.1021/acs.langmuir.2c02847
Langmuir 2023, 39, 1111−1122

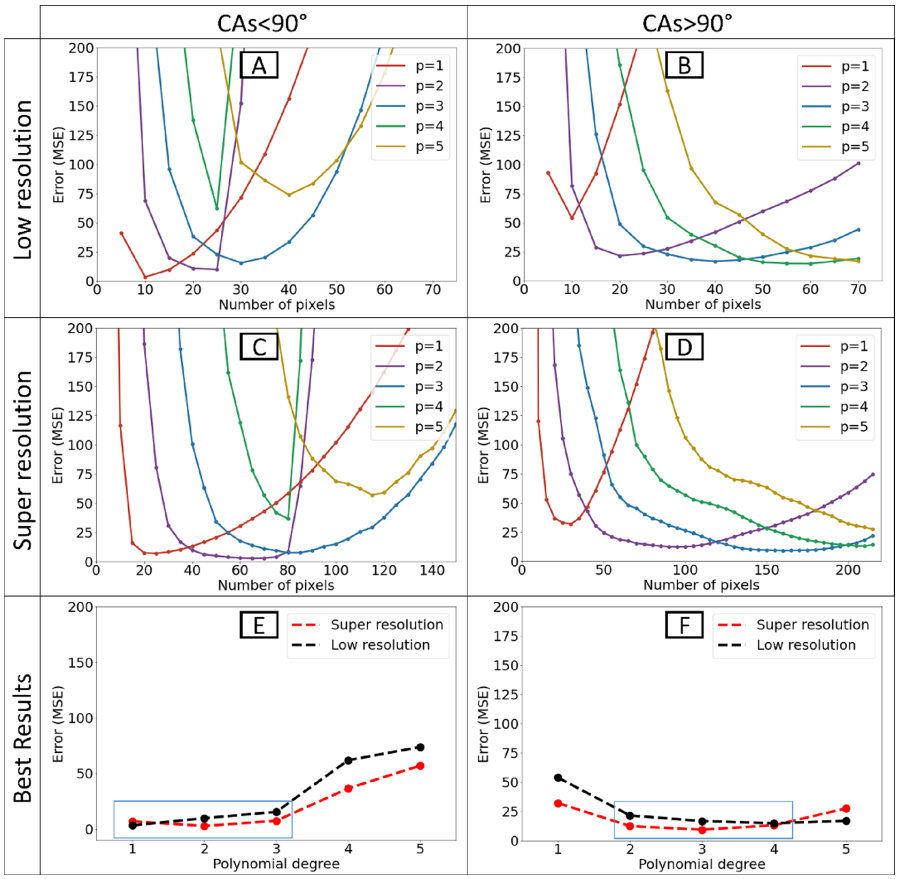


Figure 5. Measuring errors using low and super-resolutions and polynomial fitting with different parameters. CAs were divided into two categories: CA <90° and CA >90°. Based on eq 1, errors were calculated by comparing measured values with real values of the synthetic images. For clarity, we only plot values until the polynomial of an order 5. The error is based on the number of pixels and order of a polynomial for low-resolution images in the hydrophilic (A) and hydrophobic (B) parts and super-resolution images in the hydrophilic (C) and hydrophobic (D) parts. The best results were achieved by examining the different numbers of pixels based on each polynomial order for hydrophilic (E) and hydrophobic (F) parts.

samples, we obtain a minimum error of 6.28. This error value corresponds to $p = 3$ and $n = 160$ (Figure 5D). The decrease of the error from 10.9 to 6.28 indicates that the super-resolution images significantly improve the precision of the extracted CA for CAs >90°. Now, we used a binary segmentation, corresponding to CA >90° or <90°. The next step will be to investigate if a better segmentation of CAs will further decrease the error.

CA Segmentation Optimization. Based on the prior segmentation (2 segment, from 15 to 90 and from 90 to 165), the lowest error for super-resolution images for CAs <90° was 2.82 ($p = 2$ and $n = 65$; Figure 6A) and for 6.28 ($p = 3$ and $n = 160$; Figure 6B) for CAs >90°. However, the error for 15−30, 75−90, 135−150, and 150−165 exceeds the corresponding value in the respective segments (Figure 6A,B). Hence, additionally refining these CA classes into smaller segments may further improve the error.

Using a grid search algorithm,[49] we tested different segmentations. 4 segments improved the error (Figure 6C). CAs within 15−60° are measured when $p = 2$ and $n = 60$, 60−105° when $p = 2$ and $n = 85$, 105−135° when $p = 3$ and $n = 175$, and 135−165° when $p = 4$ and $n = 215$. To compare the accuracies, the black bars represent the results of the optimized polynomial on low-resolution images (Figure 6C). The green bars represent the 2-segment super-resolution optimized-fitting (2S-SROF) results, and the yellow bars represent the 4-segment super-resolution optimized-fitting (4S-SROF) results. Finally, the total error for >90° improved to 4.91 and for <90° improved to 2.11.

In conclusion, polynomial fitting without prior optimization can lead to a significant error. For example, if someone considers $p = 2$ and $n = 20$ (a rational choice) to determine CAs for <90° with low-resolution images, the accuracy based on MSE will be 10.76, which is more than 3 times bigger than the optimal value (3.41; $p = 1$, $n = 10$). The accuracy depends heavily on the selected n and p. The optimized polynomial fitting error before using super-resolution was 1.8° for CAs <90° and 3.3° for CAs >90°. After using the super-resolution procedure, the error decreased to 1.4° for CAs <90° and 2.2° for CAs >90° based on RMSE. We want to emphasize that this error calculation is based on our circular model and is only a subset of the shapes that a drop can take. Therefore, we prefer

https://doi.org/10.1021/acs.langmuir.2c02847
Langmuir 2023, 39, 1111−1122

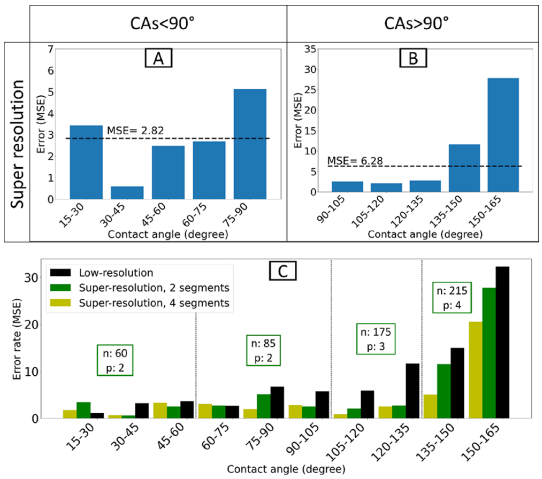


Figure 6. Distribution of error based on CAs. (A) The MSE for super-resolution images based on CAs from 15 to 90. The average MSE is 2.82 for $p = 2$ and $n = 65$. (B) The MSE for super-resolution images based on CAs from 90 to 165. The average MSE is 6.28 for $p = 3$ and $n = 160$. (C) New segmentation led to an improvement in MSE for both hydrophobic (4.91) and hydrophilic (2.11) parts. The segmentation and parameters in the figure are related to green bars. The yellow bars correspond to the results obtained with 2 segments and the black bars for the original low-resolution image for comparison.

to report the improvements by providing percentages. The accuracy improved by 21% for CAs <90° and 33% for CAs >90° when using the 4S-SROF. We assume that the analysis of images recorded in real experiments improves by a similar percentage.

We have observed that the polynomial fitting has a shortcoming to calculate the CA for a line close to the vertical line (90°). This is because finding combinations of values for polynomial fittings that may produce a vertical line is problematic. A similar problem with using polynomials has been reported in other studies.[14] In order to keep the CA accuracy near 90° for all calculations, we have rotated the drop boundary by 90° and calculated θ_a and θ_r while taking into account the rotation for all calculations.

APPLICATION

How the Toolkit Works. For our accurately known drop contours, we developed an open-source toolkit.[39] We will briefly explain how the toolkit extracts different criteria below.

Right and Left Halves' Coordinates. Consider the drop contour as two separate halves: right and left. $X_r = \{x_{r1}, ..., x_{rn}\}$ and $Y_r = \{y_{r1}, ..., y_{rn}\}$ are the coordinates of the right half of the drop. $X_l = \{x_{l1}, ..., x_{ln}\}$ and $Y_l = \{y_{l1}, ..., y_{ln}\}$ are the coordinates of the left half of the drop, and n is the number of drop contour pixels.

CAs. The first step in measuring CAs is to determine how many pixels should be selected at the front and the rear. Consider that m pixels from the right half and k pixels from the left half are needed. In this case, we will select $\{x_{r1}, ..., x_{rm}\}$ and $\{y_{r1}, ..., y_{rm}\}$ pixels from the right half and $\{x_{l1}, ..., x_{lk}\}$ and $\{y_{l1}, ..., y_{lk}\}$ from the left side. Using the extracted pixels, a polynomial fitting algorithm can be used to measure CAs.

Drop Length. The drop length depends on the difference between the two end pixels of the drop contour. The X coordinates for these two pixels are x_{l1} and x_{r1}. As a result, $|x_{r1} - x_{l1}|$ represents the length of the drop. With polynomial fitting, the accuracy of determining the length of the drop increases (Figure S11).

Median Line Angle. $M = \{\frac{x_{r1} + x_{l1}}{2}, ..., \frac{x_{rn} + x_{ln}}{2}\}$ represents the average point for each row of the drop contour. These values resemble a tilted vertical line when plotted (Figure S12). We observed that drop oscillation can be perfectly represented by the angle between the line and the horizontal line. We used its angle as a criterion. The weighted average function on M represents a number related to the x coordinate of a vertical line that can be used to divide the drop volume in half horizontally. The formula will be

$$X_{center} = \frac{\sum_i^n \left(\frac{x_{ri} + x_{li}}{2} \cdot |x_{ri} - x_{li}|\right)}{\sum_i^n (|x_{ri} - x_{li}|)} \quad (11)$$

Velocity. Previously, the average velocity of the two end pixels of the drop contour was used to calculate the velocity.[50] This method considers the middle of the drop length as the drop's center. However, it is not representative of the velocity of other parts of the drop. Drop velocity can be measured using eq 11 since it is based on the volume distribution of the drop. Using eq 11, with respect to the camera frame rate when capturing the frames, it is possible to calculate the drop speed based on the center position of the drop in each frame.

Refer to the video in the Supporting Information for a visual representation of the extracted edge and different criteria.

We applied the above toolkit to investigate a sample with a defect. We recorded high-speed videos of 35 μL drops of deionized water sliding down tilted surfaces. The tilt angle was 50°. The hydrophilic round defect had a diameter of 1100 μm. We applied optimized polynomial fits on low-resolution images and the above discussed 4S-SROF and extracted θ_a and θ_r (Figure 7). In addition, we compared the CAs with the ones calculated by a second-order polynomial fit, which is frequently reported in the literature.[14,15] Both the optimized polynomial fits and 4S-SROF lead to CAs that are 108° (Figure 7A) and 75° (Figure 7B). In addition, the 4S-SROF fits result in reduced fluctuations of the signals. The limited number of available data points (n) in low-resolution images causes the fluctuations. Here, super-resolution reduces fluctuations and thus increases accuracy due to more available data points. As a result, the measurement of CAs >90° is greatly improved by the super-resolution method (Figure 7A). In the 4S-SROF, the trend will be more obvious. It is due to improved accuracy when calculating super-resolution images (Figure 6C). The accuracy of CAs depends on the initial variables as well as the amount of noise in the recorded video. We discuss how a negative situation can affect CAs' accuracy in the sensitivity analysis section in the Supporting Information.

Samples with a Chemical Heterogeneity. We will plot and discuss the drop's movement on different samples in this and the next section. As the toolkit output, we are interested in the details of movement rather than its physics, which would require more experiments with different parameters. To analyze the sample with a chemical heterogeneity, we selected a tilt angle of 35° (Figure 8A_i). The tilt angle was not high enough for drops to slide on the POS surface. To accelerate the drops reaching the OTS surface, the needle and surface

https://doi.org/10.1021/acs.langmuir.2c02847
Langmuir 2023, 39, 1111−1122

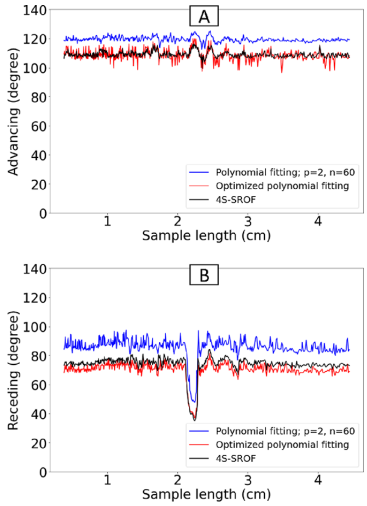


Figure 7. Comparison of a second-order polynomial fitting with an optimized polynomial fitting as well as a 4S-SROF for both advancing angle (A) and receding angle (B). The blue line is polynomial fitting without optimization on low-resolution images for $p = 2$ and $n = 60$. The red line is polynomial fitting after optimization on low-resolution images for $p = 3$ and $n = 40$. The black line is polynomial fitting after optimization on super-resolution images (4S-SROF).

distance was increased to 0.5 cm. On OTS, the CA hysteresis is lower and the drop accelerates. When the advancing part reached the first transition line ($x = 1.4$ cm), θ_a dropped by 7° (Figure 8A_ii), and when it reached the second transition line ($x = 2.7$ cm), θ_a increased by 5°. The θ_r did not change in $x = 1.4$ cm, neither $x = 2.7$ cm. However, θ_r increased and decreased at $x = 1.9$ cm and $x = 3.2$ cm where the receding part touched the transition lines, respectively. The exact position of the first and second transition lines can be determined precisely by using the changing points of θ_a and θ_r, considering the drop length information for each moment of sliding. Therefore, $x = 1.64$ cm and $x = 2.93$ cm are the first and second transition lines' positions, respectively (Figure 8A_iii–v, red lines). The blue and red areas represent when the drop is completely in POS and OTS, respectively. Observing the jump of the θ_a and θ_r when they cross the transition line was more clear when we conducted three independent experiments (in Figure 8A_ii, three lines for θ_a and θ_r are plotted). A smooth change in θ_a and its inherent difficulty in measuring CAs beyond 90° justify using the proposed method for analyzing θ_a. Based on standard dynamic CA measurement, we measured $\theta_a = 124°$ and $\theta_r = 83°$ for POS and $\theta_a = 113°$ and $\theta_r = 92°$ for OTS. Based on sliding drop measurement, θ_a and θ_r values behave similarly to standard measurements. There is a decrease in their values due to the velocity, especially at the end of the movement. Due to the distance between the needle and the surface, the drop length at the starting point fluctuates rapidly (Figure 8A_iii). In general, the drop length on OTS is less than POS. The drop moves slower on the POS than on the OTS when it is on the first transition line (POS to OTS). In the same way, the drop length decreases as the drop passes the second transition line. The velocity of the drop decreases on POS and increases on OTS (Figure 8A_iv). In this way, the effect of changing the surface on velocity differences can be seen clearly. As a result of needle distance and two transition lines, the drop fluctuates throughout the route (Figure 8A_v). Water drop oscillations are clearly seen to increase after passing through transition lines.

Samples with a Topographic Defect. For the sample with defect (Figure 8B_i), we calculated the drop's CAs of $\theta_a = 108°$ and $\theta_r = 75°$ (Figure 8B_ii). At first, θ_a plateaued with an average of 108° and a standard deviation of 2°. Similarly, θ_r had an average of 75°. When the center of the drop reached a sliding length of $x = 1.75$ cm (1st blue vertical line). At this position, the advancing side of the drop touched the defect and θ_a dropped from 108° to 101°. Then, the drop's receding side touched the defect at a drop's center position of $x = 2.11$ cm (2nd blue vertical line) and θ_r decreased from 75° to a minimum value of 39°. Then, the three-phase contact line of the drop depinned at $x = 2.27$ cm (3rd blue vertical line). Now, θ_r reached its previous value and remained almost constant with a regular oscillation. The blue color area indicates that the defect is located inside the drop, and the red color area signifies that the drop is pinned. Before touching the defect, the drop length diagram was plateaued (Figure 8B_iii). After the advancing part touched the defect, the drop length increased rapidly and plateaued at a higher value when the defect was inside the drop (blue color area). While the receding part got stuck when it touched the defect, the advancing part kept moving. Therefore, the drop length started to increase gradually (red color area) and, after depinning, the drop length oscillated regularly. The velocity of the drop was not sensitive to the advancing part but when receding touched the defect velocity decreased significantly (Figure 8B_iv). The slope of the line became the same as before touching the defect after passing the defect. The median line angle was quite constant before touching the defect due to the gentle placement of the water drop (there was no needle distance). When the advancing contact lines touched the defect, the median line angle started to fluctuate slightly. Once the receding part depinned from the defect, we measured stronger fluctuations in drop movement (Figure 8B_v). These fluctuations correspond to drop oscillations, which are excited by the interaction with the defect.

To analyze drop motion, we calculated the potential and kinetic energies of a drop:

$$E_g = mgh \tag{12}$$

$$E_k = \frac{1}{2} mv^2 \tag{13}$$

$$E_g = E_k + E_r + E_d \tag{14}$$

where E_g is the potential energy, E_k is the kinetic energy, E_r is the rolling energy, and E_d is the dissipated energy.

The kinetic energy values have been smoothed by using a Savitzky–Golay filter[51] (Figure 9A, black dots). It started from 46 nJ and increased up to 201 nJ. The kinetic energy before and after touching the defect was fitted with a second-order polynomial (due to the accelerated movement; Figure 9A, blue lines). The fitting error is about 7 nJ. Based on the difference between the two fitted diagrams (red line length), we calculated the dissipation energy due to the defect to be 68 nJ. Thus, the total energy changes in the kinetic energy diagram from beginning to end are 154 nJ, and the energy dissipation due to the defect is 68 nJ or 44%.

https://doi.org/10.1021/acs.langmuir.2c02847
Langmuir 2023, 39, 1111−1122

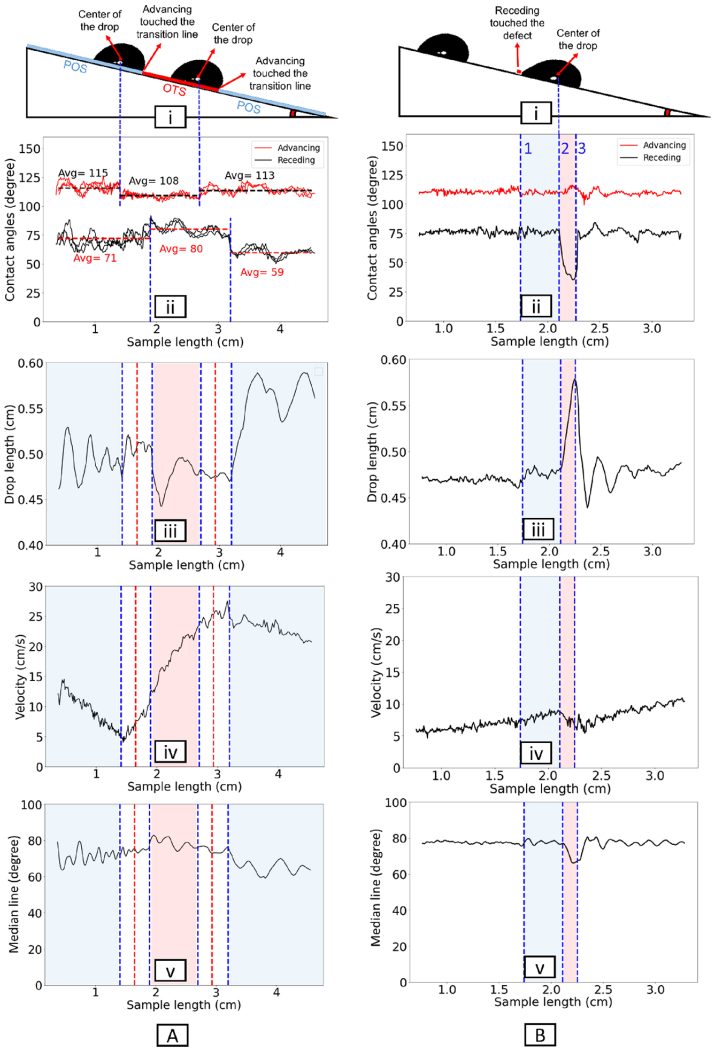


Figure 8. The drop profile when it is sliding on a sample with a defect and a sample with a chemical heterogeneity. Column A; A_i: The sliding drop on a sample with a chemical heterogeneity. A_ii: The CAs' diagram based on the sample length. The first and third blue lines indicate when the advancing part touched the new surface, and the second and fourth blue lines indicate when the receding part touched the new surface. The red line represents the exact position of the transition line calculated from the blue lines and drop length. The blue area represents when the drop is completely on the POS, while the red area represents when it is completely on the OTS. A_iii: The receding diagram based on the sample length. A_iv: The velocity diagram based on the sample length. A_v: The median line angle based on the sample length. Column B; B_i: The sliding drop on a sample with a defect. B_ii: The CAs' diagram based on the sample length. The first blue line represents the advancing part touching the defect, the second blue line represents the receding part touching and pinning the defect, and the third blue line represents the receding part depinning from the defect. There is a blue area when the defect is inside the drop and a red area when the drop is pinned. B_iii: The drop length diagram based on the sample length. B_iv: The velocity diagram based on the sample length. B_v: The median line angle based on the sample length.

To compare potential and kinetic energy changes, potential energy is also calculated. The potential energy was 6.6 μJ at the beginning and decreased linearly (based on the sample length; Figure 9B). Kinetic energy changes account for only 2.3% of potential energy changes. Low velocity (high surface energy) on the surface with a defect causes this. It means that 97.7% of the potential energy dissipated or partially converts to the rolling energy based on eq 14.

The accuracy of the described criteria depends on initial variables and existing noises in the captured video. By conducting a sensitivity analysis, it is possible to determine how much a negative situation affects accuracy. The sensitivity analysis is discussed in the Supporting Information, in particular, how different criteria are affected by noises, noise removal algorithms, baseline location errors, and tilt angle measurement errors. Based on sensitivity analysis, we determined that the baseline position is the most crucial

https://doi.org/10.1021/acs.langmuir.2c02847
Langmuir 2023, 39, 1111−1122

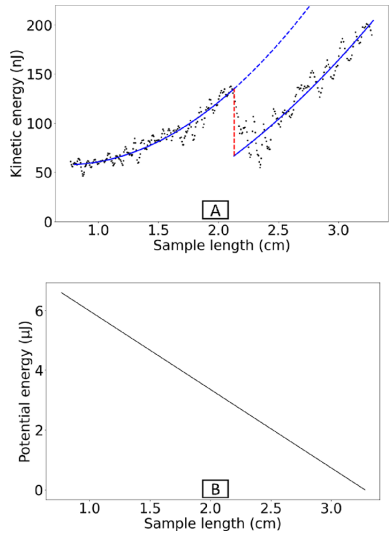


Figure 9. The kinetic and potential energy of the surface with the defect. (A) For each moment of movement, the black dots represent the kinetic energy. The blue lines represent the second-order polynomials fitted to the black dots. There is a difference between the two blue lines representing the defect's effect on the energy dissipation that is marked by the red line. (B) Potential energy based on sample length.

parameter for obtaining accurate CAs. Edge detection algorithms have difficulty detecting the transition line between the real drop and its reflection when the surface is transparent. The baseline detection in the transparent samples section in the Supporting Information introduces a method independent of edge detection to detect baseline in transparent samples.

SUMMARY AND CONCLUSIONS

In order to optimize polynomial fitting and measure the accuracy, we used synthesized images. The polynomial parameters were adjusted separately for the front and rear of the drop. By finding the best values for the n and p according to the sliding drop image resolution, the accuracy based on MSE is 3.41 for angles 15−90° and 10.9 for angles 90−165°. A super-resolution model based on deep learning was developed that increased the original image 9 times with an accuracy of 36.39 PSNR. Magnified images improved the measurement accuracy as discussed. A final improvement arose from the fact that the error distribution is not the same for different angles. As a result, the angle measurement accuracy was once again improved to 2.11 for CAs <90° and to 4.91 for CAs >90°. This means that the accuracy improved by 21% for CAs <90° and 33% for CAs >90° when using a 4S-SROF.

We developed a toolkit that can automatically extract the drop profile at every moment of movement. The parameters extracted are the drop length, middle line angle, velocity, receding angle, and advancing angle. These parameters can be extracted simultaneously to help researchers conduct detailed studies with a broad range of variables taking into consideration their correlations.

ASSOCIATED CONTENT

Supporting Information

The Supporting Information is available free of charge at https://pubs.acs.org/doi/10.1021/acs.langmuir.2c02847.

Supporting figures, sensitivity analysis, and baseline detection for special cases (PDF)

Video showing how the toolkit works, an example of drop sliding on a sample with a topographic defect (MP4)

AUTHOR INFORMATION

Corresponding Author

Rüdiger Berger − *Max Planck Institute for Polymer Research, D-55128 Mainz, Germany;* orcid.org/0000-0002-4084-0675; Email: berger@mpip-mainz.mpg.de

Authors

Sajjad Shumaly − *Max Planck Institute for Polymer Research, D-55128 Mainz, Germany*

Fahimeh Darvish − *Max Planck Institute for Polymer Research, D-55128 Mainz, Germany*

Xiaomei Li − *Max Planck Institute for Polymer Research, D-55128 Mainz, Germany*

Alexander Saal − *Max Planck Institute for Polymer Research, D-55128 Mainz, Germany*

Chirag Hinduja − *Max Planck Institute for Polymer Research, D-55128 Mainz, Germany;* orcid.org/0000-0002-1047-5750

Werner Steffen − *Max Planck Institute for Polymer Research, D-55128 Mainz, Germany;* orcid.org/0000-0001-6540-0660

Oleksandra Kukharenko − *Max Planck Institute for Polymer Research, D-55128 Mainz, Germany;* orcid.org/0000-0002-3285-1403

Hans-Jürgen Butt − *Max Planck Institute for Polymer Research, D-55128 Mainz, Germany;* orcid.org/0000-0001-5391-2618

Complete contact information is available at: https://pubs.acs.org/10.1021/acs.langmuir.2c02847

Author Contributions

S.S. carried out deep model, toolkit development, and coding. F.D. designed the synthetic images experiment. S.S. and F.D. performed experiments and analyzed data. X.L. did experiments and prepared the dataset. A.S. and C.C. prepared samples with a topographic defect and with chemical heterogeneity. S.S., F.D., and R.B. wrote the manuscript. S.S., R.B., H.-J.B., O.K., and W.S. discussed the results regularly.

Funding

Open access funded by Max Planck Society.

Notes

The authors declare no competing financial interest.

ACKNOWLEDGMENTS

We thank Andreas Best for measuring the sphere diameter and Joseph Rudzinski for discussions. We acknowledge the financial support from Max Planck Center on Complex Fluid Dynamics (S.S.) and the Priority Programme 2171 Dynamic wetting of flexible, adaptive, and switchable surfaces (grant nos. BU 1556/36 and BE 3286/6-1: H.-J.B., R.B., and X.L.). We acknowledge the financial support by the German Research

https://doi.org/10.1021/acs.langmuir.2c02847
Langmuir 2023, 39, 1111−1122

Society via the CRC 1194 (Project-ID 265191195) "Interaction between Transport and Wetting Processes", projects C07N (C.H., R.B., H.-J.B., and A.S.). This project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement no. 883631) (F.D. and H.-J.B.).

REFERENCES

(1) Bormashenko, E. Wetting of real solid surfaces: new glance on well-known problems. *Colloid Polym. Sci.* **2013**, *291*, 339−342.
(2) Young, T., III An essay on the cohesion of fluids. *Philos. Trans. R. Soc. London* **1805**, *95*, 65−87.
(3) Liu, K.; Vuckovac, M.; Latikka, M.; Huhtamäki, T.; Ras, R. H. A. Improving surface-wetting characterization. *Science* **2019**, *363*, 1147−1148.
(4) Hartland, S. *Surface and interfacial tension: measurement, theory, and applications*; CRC Press: 2004, DOI: 10.1201/9780203021262.
(5) Rotenberg, Y.; Boruvka, L.; Neumann, A. W. Determination of surface tension and contact angle from the shapes of axisymmetric fluid interfaces. *J. Colloid Interface Sci.* **1983**, *93*, 169−183.
(6) del Río, O. I.; Neumann, A. W. Axisymmetric Drop Shape Analysis: Computational Methods for the Measurement of Interfacial Properties from the Shape and Dimensions of Pendant and Sessile Drops. *J. Colloid Interface Sci.* **1997**, *196*, 136−147.
(7) Lamour, G.; Hamraoui, A.; Buvailo, A.; Xing, Y.; Keuleyan, S.; Prakash, V.; et al. Contact angle measurements using a simplified experimental setup. *J. Chem. Educ.* **2010**, *87*, 1403−1407.
(8) Butt, H.-J.; Liu, J.; Koynov, K.; Straub, B.; Hinduja, C.; Roismann, I.; et al. Contact angle hysteresis. *Curr. Opin. Colloid Interface Sci.* **2022**, 101574.
(9) Huhtamäki, T.; Tian, X.; Korhonen, J. T.; Ras, R. H. A. Surface-wetting characterization using contact-angle measurements. *Nat. Protoc.* **2018**, *13*, 1521−1538.
(10) Kwok, D. Y.; Neumann, A. W. Contact angle measurement and contact angle interpretation. *Adv. Colloid Interface Sci.* **1999**, *81*, 167−249.
(11) Yarin, A. L. Drop impact dynamics: splashing, spreading, receding, bouncing... *Annu. Rev. Fluid Mech.* **2006**, *38*, 159−192.
(12) Good, R. J. Contact angle, wetting, and adhesion: a critical review. *J. Adhes. Sci. Technol.* **1992**, *6*, 1269−1302.
(13) Bishop, E. A generalization of the Stone-Weierstrass theorem. *Pac. J. Appl. Math.* **1961**, *11*, 777−783.
(14) Chini, S. F.; Amirfazli, A. A method for measuring contact angle of asymmetric and symmetric drops. *Colloids Surf., A* **2011**, *388*, 29−37.
(15) Quetzeri-Santiago, M. A.; Castrejón-Pita, J. R.; Castrejón-Pita, A. A. On the analysis of the contact angle for impacting droplets using a polynomial fitting approach. *Exp. Fluids* **2020**, *61*, 143.
(16) Atefi, E.; Mann, J. A., Jr.; Tavana, H. A Robust Polynomial Fitting Approach for Contact Angle Measurements. *Langmuir* **2013**, *29*, 5677−5688.
(17) Bateni, A.; Susnar, S. S.; Amirfazli, A.; Neumann, A. W. A high-accuracy polynomial fitting approach to determine contact angles. *Colloids Surf., A* **2003**, *219*, 215−231.
(18) Kumar, A. Super-Resolution with Deep Learning Techniques: A Review. *Computational Intelligence Methods for Super-Resolution in Image Processing Applications.*; Springer: 2021:43−59, DOI: 10.1007/978-3-030-67921-7_3.
(19) Yang, W.; Zhang, X.; Tian, Y.; Wang, W.; Xue, J.-H.; Liao, Q. Deep learning for single image super-resolution: A brief review. *IEEE Trans. Multimedia.* **2019**, *21*, 3106−3121.
(20) Keys, R. Cubic convolution interpolation for digital image processing. *IEEE Trans. Acoust.* **1981**, *29*, 1153−1160.
(21) Sun, J; Xu, Z; Shum, H-Y, editors. Image super-resolution using gradient profile prior. *2008 IEEE Conference on Computer Vision and Pattern Recognition*; 2008: IEEE.
(22) Dai, S.; Han, M.; Xu, W.; Wu, Y.; Gong, Y.; Katsaggelos, A. K. Softcuts: a soft edge smoothness prior for color image super-resolution. *IEEE Trans. Image Process.* **2009**, *18*, 969−981.
(23) LeCun, Y.; Bengio, Y.; Hinton, G. Deep learning. *Nature.* **2015**, *521*, 436−444.
(24) Zhang, S.; Liang, G.; Pan, S.; Zheng, L. A fast medical image super resolution method based on deep learning network. *IEEE Access.* **2019**, *7*, 12319−12327.
(25) Wang, H.; Rivenson, Y.; Jin, Y.; Wei, Z.; Gao, R.; Günaydın, H.; et al. Deep learning enables cross-modality super-resolution in fluorescence microscopy. *Nat. Methods* **2019**, *16*, 103−110.
(26) Belthangady, C.; Royer, L. A. Applications, promises, and pitfalls of deep learning for fluorescence image reconstruction. *Nat. Methods* **2019**, *16*, 1215−1225.
(27) Liu, Y.; Sun, Q.; Lu, W.; Wang, H.; Sun, Y.; Wang, Z.; et al. General resolution enhancement method in atomic force microscopy using deep learning. *Adv. Theory Simul.* **2019**, *2*, 1800137.
(28) Hagita, K.; Higuchi, T.; Jinnai, H. Super-resolution for *asymmetric resolution* of FIB-SEM 3D imaging using AI with deep learning. *Sci. Rep.* **2018**, *8*, 5877.
(29) Liu, B.; Tang, J.; Huang, H.; Lu, X.-Y. Deep learning methods for super-resolution reconstruction of turbulent flows. *Phys. Fluids* **2020**, *32*, No. 025105.
(30) Bode, M.; Gauding, M.; Lian, Z.; Denker, D.; Davidovic, M.; Kleinheinz, K.; et al. Using physics-informed enhanced super-resolution generative adversarial networks for subfilter modeling in turbulent reactive flows. *Proc. Combust. Inst.* **2021**, *38*, 2617−2625.
(31) Dong, C.; Loy, C. C.; He, K.; Tang, X. Image super-resolution using deep convolutional networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2016**, *38*, 295−307.
(32) Dong, C; Loy, CC; Tang, X, editors. Accelerating the super-resolution convolutional neural network. *European conference on computer vision*; 2016: Springer.
(33) Kim, J; Lee, JK; Lee, KM, editors. Accurate image super-resolution using very deep convolutional networks. *Proceedings of the IEEE conference on computer vision and pattern recognition*; IEEE 2016.
(34) Kim, J; Lee, JK; Lee, KM, editors. Deeply-recursive convolutional network for image super-resolution. *Proceedings of the IEEE conference on computer vision and pattern recognition*; IEEE 2016.
(35) Shi, W; Caballero, J; Huszár, F; Totz, J; Aitken, AP; Bishop, R, et al., editors. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. *Proceedings of the IEEE conference on computer vision and pattern recognition*; IEEE 2016.
(36) Saal, A.; Straub, B. B.; Butt, H.-J.; Berger, R. Pinning forces of sliding drops at defects. *Europhys. Lett.* **2022**, *139*, 47001.
(37) Hinduja, C.; Laroche, A.; Shumaly, S.; Wang, Y.; Vollmer, D.; Butt, H.-J.; et al. Scanning Drop Friction Force Microscopy. *Langmuir* **2022**, *38*, 14635−14643.
(38) Li, X.; Bista, P.; Stetten, A. Z.; Bonart, H.; Schür, M. T.; Hardt, S.; Bodziony, F.; Marschall, H.; Saal, A.; Deng, X.; Berger, R.; Weber, S. A. L.; Butt, H. J. Spontaneous charging affects the motion of sliding drops. *Nat. Phys.* **2022**, *18*, 713−719.
(39) Shumaly, S. & others, 2022. *4S-SROF*. Available at: https://github.com/AK-Berger/4S-SROF
(40) Chollet, F. & others, 2015. *Keras*. Available at: https://github.com/fchollet/keras
(41) Abadi, M; Agarwal, A; Barham, P; Brevdo, E; Chen, Z; Citro, C, et al. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. arXiv preprint arXiv:160304467. Cornell University 2016.
(42) Hore, A; Ziou, D, editors. Image quality metrics: PSNR vs. SSIM. *2010 20th international conference on pattern recognition*; 2010: IEEE.
(43) Canny, J. A computational approach to edge detection. *IEEE Trans. Pattern Anal. Mach. Intell.* **1986**, *PAMI-8*, 679−698.
(44) Bradski, G. The openCV library. *Dr Dobb's Journal: Software Tools for the Professional Programmer.*; M&T Pub. 2000;25(11):120−3.

https://doi.org/10.1021/acs.langmuir.2c02847
Langmuir 2023, 39, 1111−1122

(45) Decker, E. L.; Garoff, S. Contact Line Structure and Dynamics on Surfaces with Contact Angle Hysteresis. *Langmuir* **1997**, *13*, 6321−6332.
(46) Kalantarian, A.; David, R.; Neumann, A. W. Methodology for High Accuracy Contact Angle Measurement. *Langmuir* **2009**, *25*, 14146−14154.
(47) Xu, Z.; Wang, S. Y. A highly accurate dynamic contact angle algorithm for drops on inclined surface based on ellipse-fitting. *Rev. Sci. Instrum.* **2015**, *86*, No. 025104.
(48) Park, J.; Kumar, S. Droplet Sliding on an Inclined Substrate with a Topographical Defect. *Langmuir* **2017**, *33*, 7352−7363.
(49) Liashchynskyi, P.; Liashchynskyi, P.. Grid search, random search, genetic algorithm: a big comparison for NAS. arXiv preprint arXiv:191206059. Cornell University 2019.
(50) Andersen, N. K.; Taboryski, R. Drop shape analysis for determination of dynamic contact angles by double sided elliptical fitting method. *Meas. Sci. Technol.* **2017**, *28*, No. 047003.
(51) Press, W. H.; Teukolsky, S. A. Savitzky-Golay smoothing filters. *Comput. Phys.* **1990**, *4*, 669−672.

https://doi.org/10.1021/acs.langmuir.2c02847
Langmuir 2023, 39, 1111−1122

3.2 Supporting information

Supporting information

Deep learning to analyze sliding drops

Sajjad Shumaly†, Fahimeh Darvish†, Xiaomei Li†, Alexander Saal†, Chirag Hinduja†,

Werner Steffen†, Oleksandra Kukharenko†, Hans-Jürgen Butt†, Rüdiger Berger†*

† Max Planck Institute for Polymer Research, Ackermannweg 10, D-55128, Mainz, Germany

* Corresponding Author. Email: berger@mpip-mainz.mpg.de

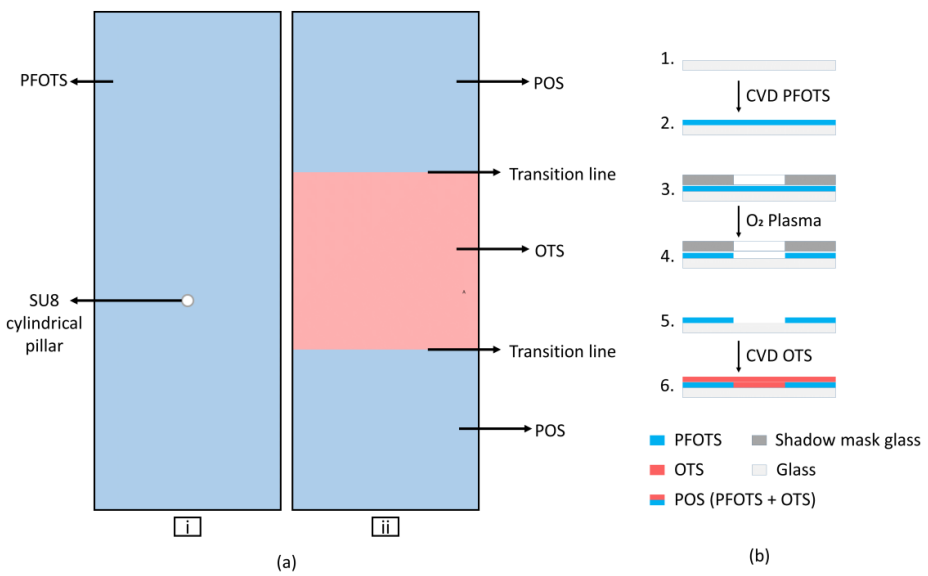


Figure S1. A schematic showing the appearance of samples. a_i: a sample with a topographic defect is covered by PFOTS and a SU8 cylindrical pillar is in the middle of the sample. a_ii: a sample with a chemical heterogeneity contains two different areas (OTS and POS) and two different transition lines (from POS to OTS and from OTS to POS). b: steps of preparing a sample with a chemical heterogeneity are represented for a better understanding.

S2

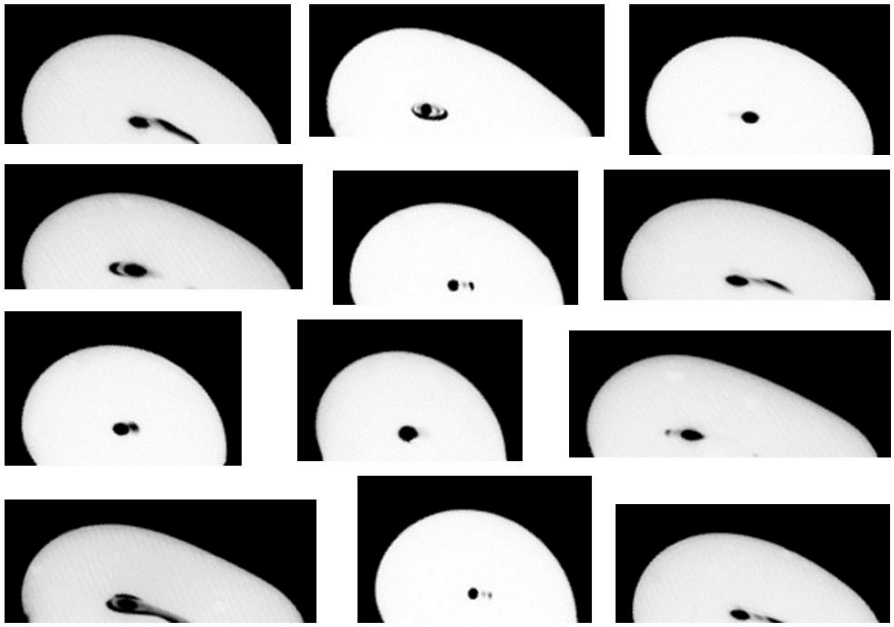


Figure S2. Examples of drop images in the dataset used for model training.

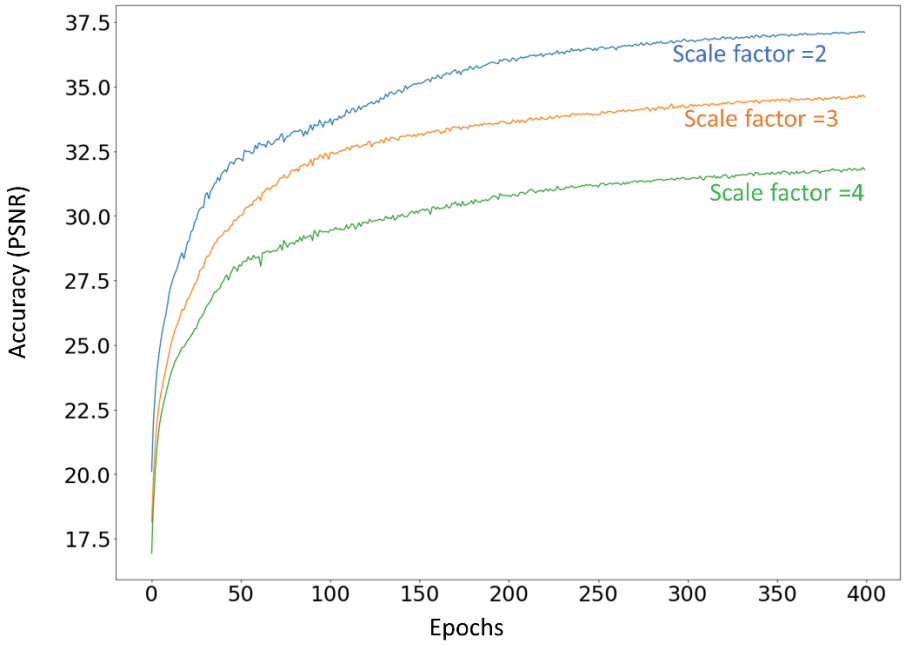


Figure S3. Effect of changing scale factor on accuracy diagram. We considered 50 drop images from the test set to compare these three diagrams in each epoch. The figure shows as the scale factor increases, the problem becomes more complex and accuracy decreases.

S4

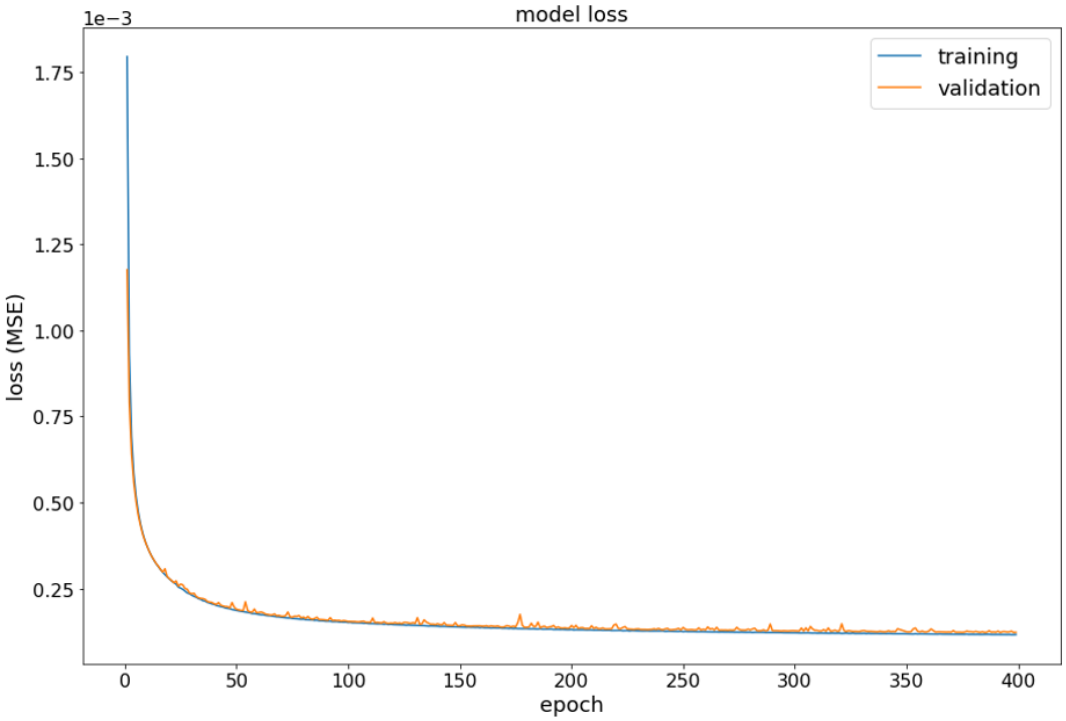


Figure S4. The training process of the modified ESPCN model based on MSE as the loss function.

S5

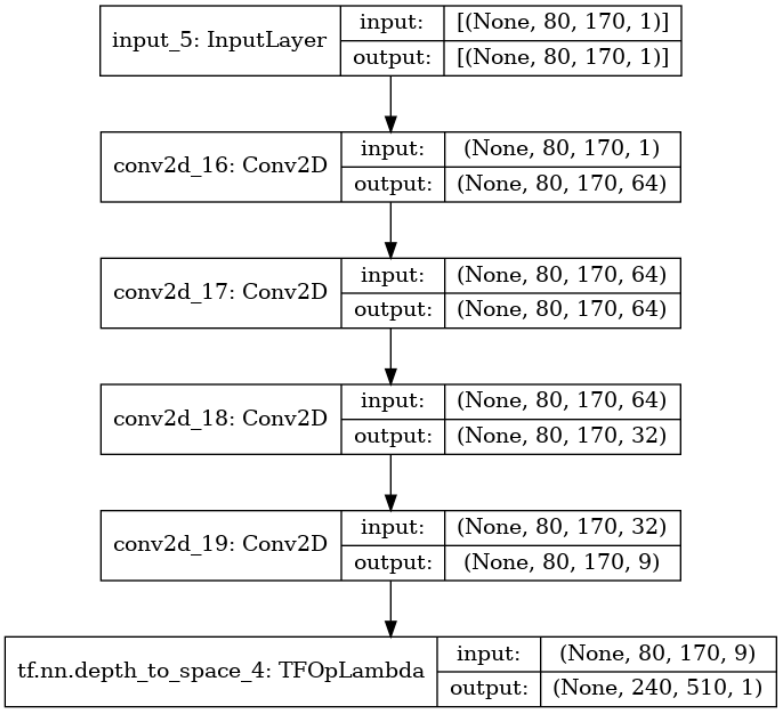


Figure S5. The modified ESPCN architecture.

S6

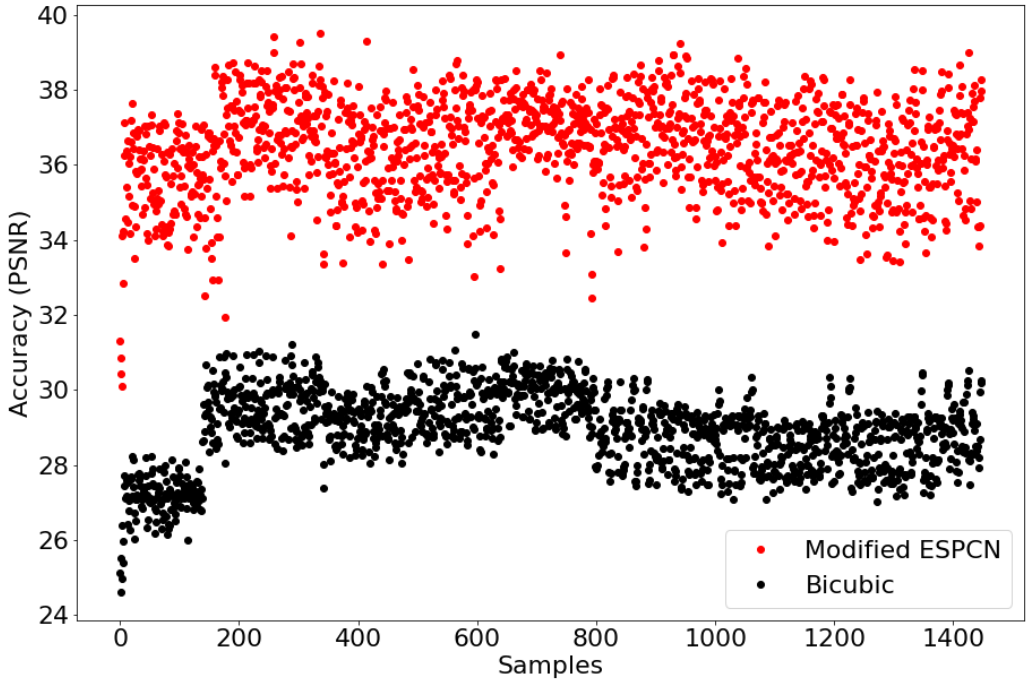


Figure S6. . Accuracy distribution of the Bicubic method and the modified ESPCN for the same 1400 samples of the test data set. All test samples belong to separate videos that were not used in the training process. The average PSNR for the Bicubic method corresponds to 28.90 and for the modified ESPCN to 36.39.

S7

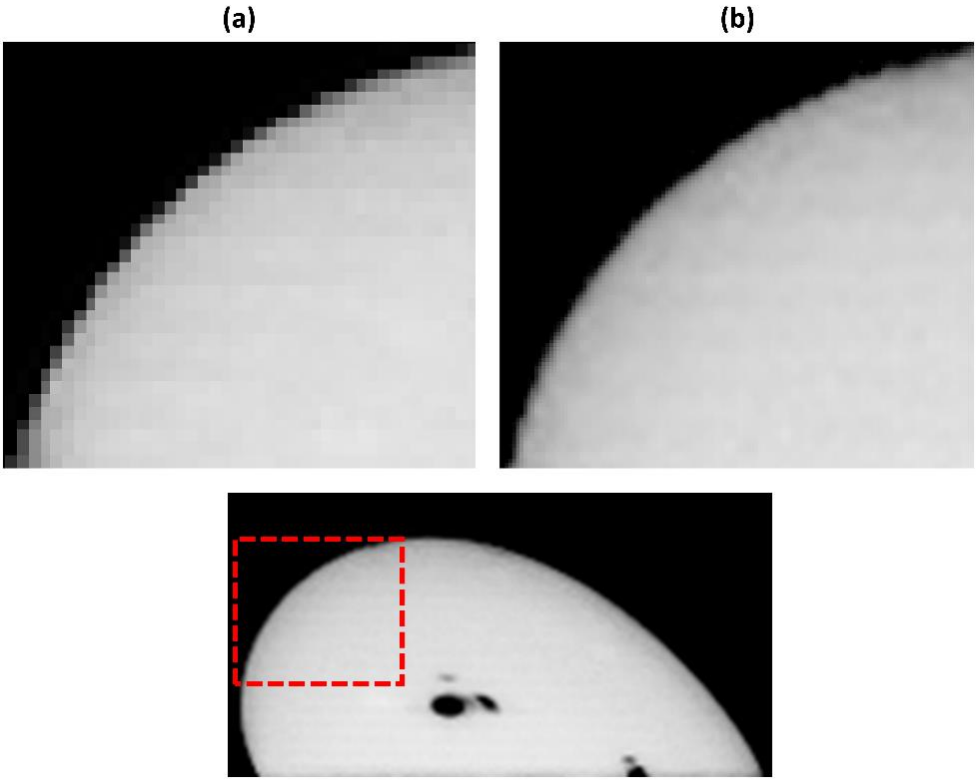


Figure S7. The accuracy of drop images before (a) and after (b) the super-resolution model.

S8

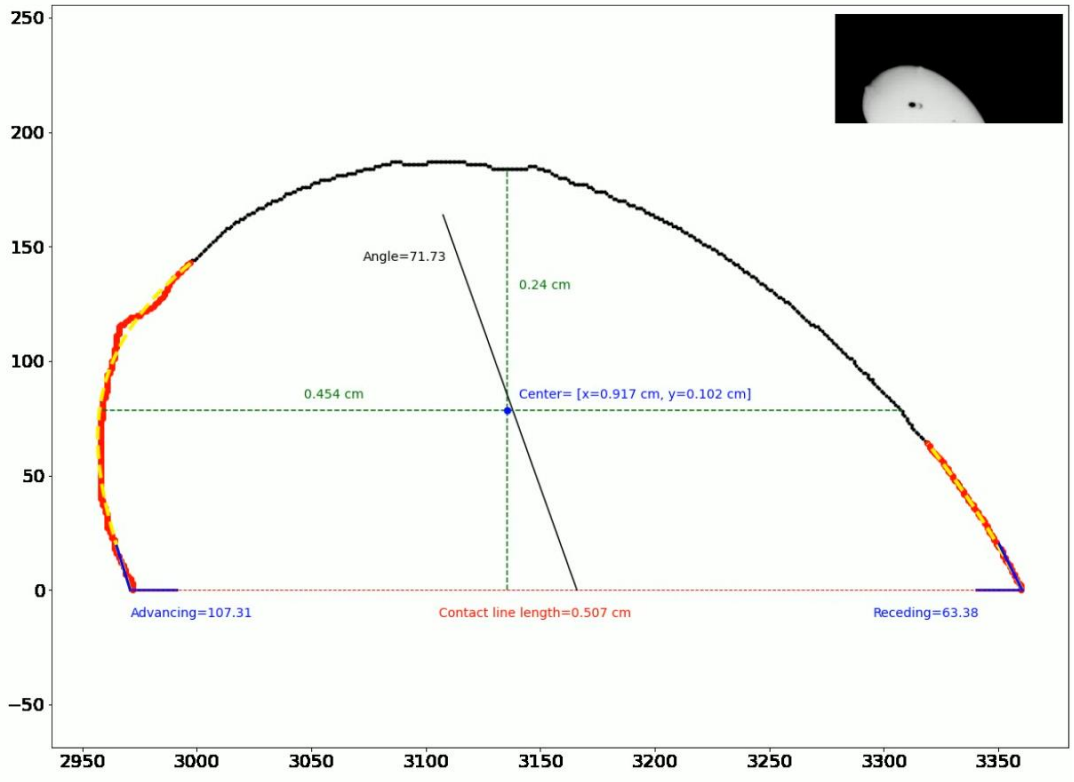


Figure S8. In the presence of noise, third-order polynomials can be properly generalized. Red dots represent selected pixels as input for polynomial fitting and yellow dashed represents third-order fitted polynomials. The drop edge is partially distorted but the polynomial fitting is mimicking the curve of the drop correctly.

S9

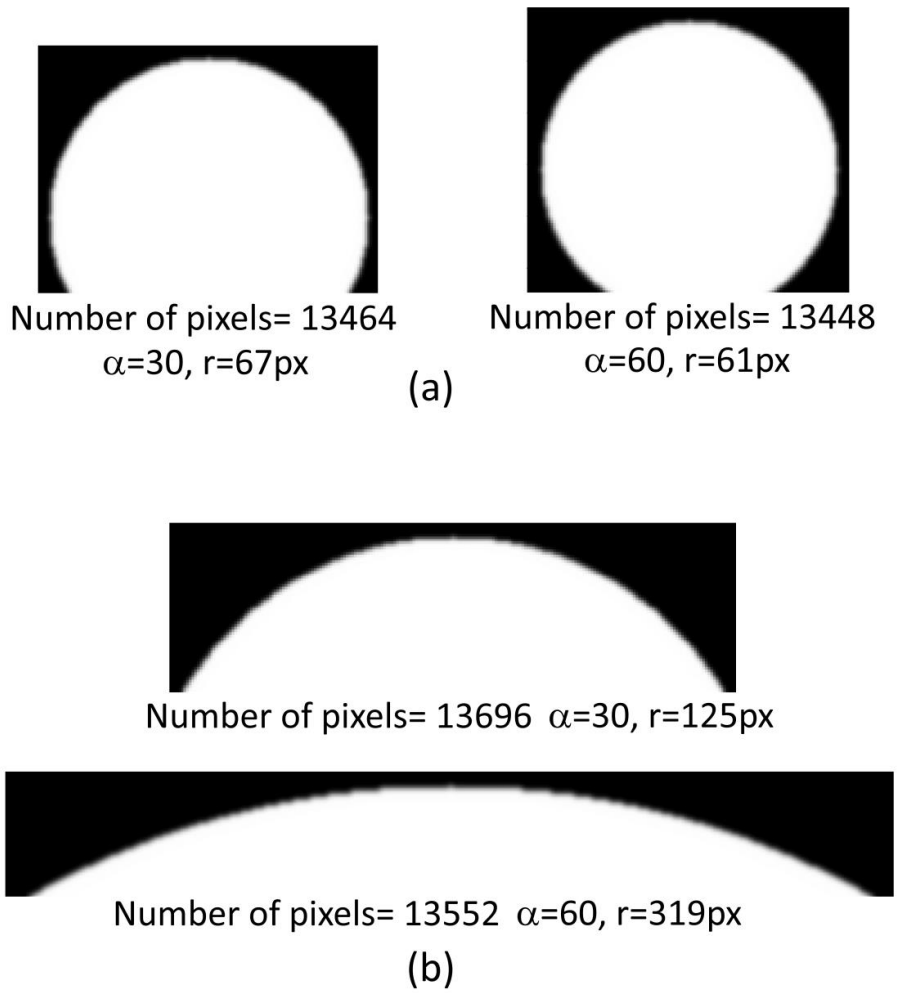


Figure S9. Some examples of generated drop images when α is 30, and 60 for >90° (a) and <90° (b).

S10

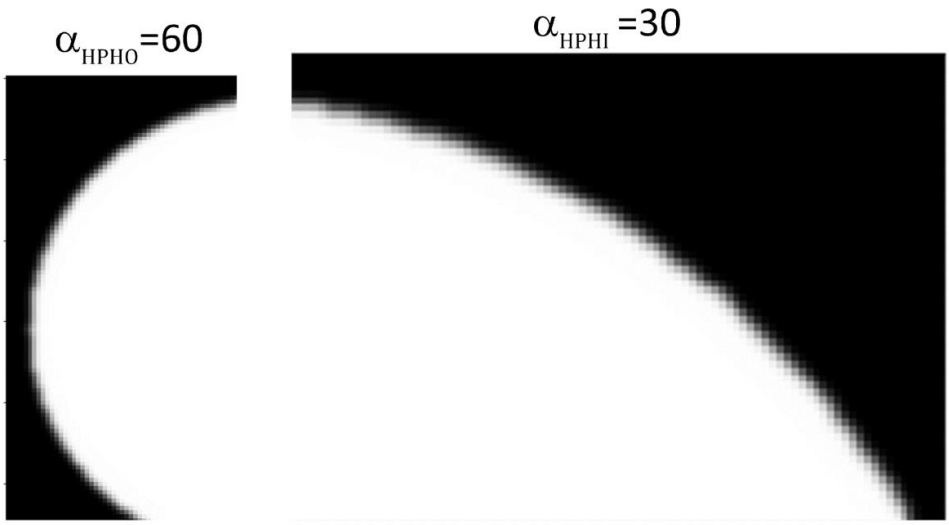


Figure S10. Combination of hydrophobic with α = 60 and hydrophilic with α = 30 synthetic images to simulate a sliding drop advancing part and receding part.

S11

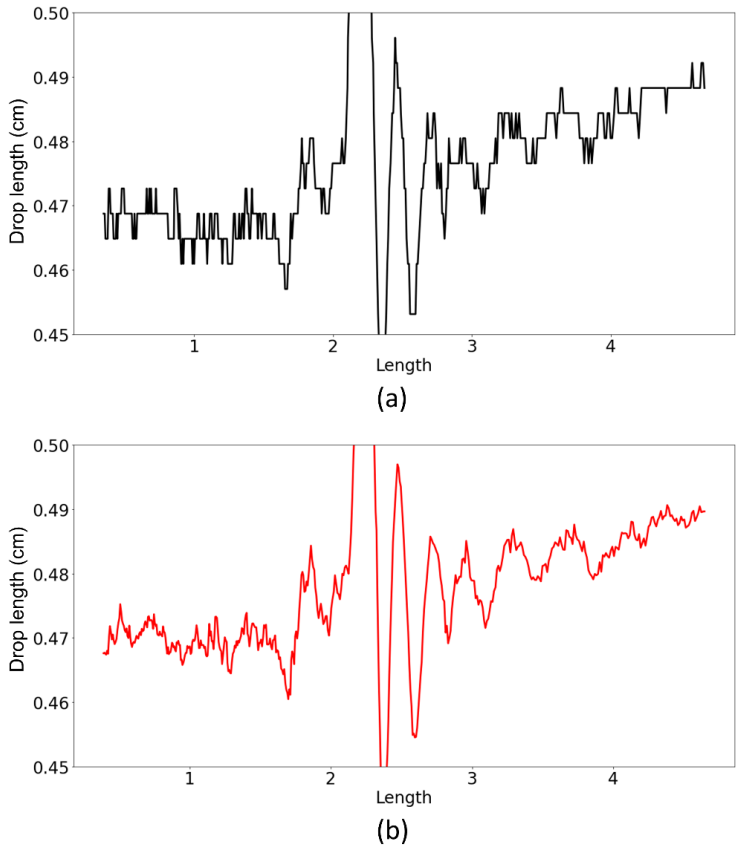


Figure S11. Changes in the drop length (cm) of a drop on a sample with a defect based on sample length (cm). Calculated drop length a) before using the polynomial fitting on a low-resolution image b) after using the polynomial fitting on a super-resolution image. The drop length depends on the difference between the two end pixels of the drop curve. But using the polynomial fitting, the exact location of the two end pixels of the drop curve is determined by its adjacent pixels. This approach increases accuracy.

S12

Figure S12. A drop contour in super-resolution space. Red dots represent the middle of each row of the image. Their shape resembles a line. The green line is a fitted line to them. This line is considered the median line. The blue line is the weighted average of the red dots. This line is used to determine the position of the drop on the sample length and velocity calculations are based on the blue line as well.

Sensitivity Analysis

Existing noises in sliding drop videos are a serious problem that needs to be addressed. Most of the noises belong to the background of the video and removing them is not challenging. But some others may appear due to different reasons. Using noise removal algorithms can cause unwanted side effects on drop boundaries. The other issue is that drop profile extracting can be affected by errors in initial variables, such as baseline location and tilt angle. This section discusses the sensitivity of the sliding drop problem to noises, noise removal algorithms, baseline location errors, and tilt angle measurement errors.

Noise handling

Noise can sometimes be distinguished by drop boundaries. Some techniques like median filters can be used to remove noise in this situation. But, noises may be very close to drop boundaries and difficult to detect. The maximum effect they have on CAs values will be when they are close to the baseline. An advancing diagram related to a sliding drop video may be influenced significantly by noises (Figure S13). Figure S13_1 represents a normal sliding drop image without any noises. But sometimes there are noises inside the taken image. In some cases, the noise is close to the drop edge (Figure S13a_2). Removing this type of noise is not possible and drop boundaries will be damaged. In this case, noise is a part of the drop and distinguishing them is not possible based on image processing algorithms. But, after extracting the drop profile and contact angles, a big leap represents noise that can be removed using signal processing techniques. In other cases where the noise has not disturbed the edge of the drop, it can be removed by image processing methods (Figure S13a_3).

We studied noise removal algorithms' effects on an advancing diagram (Figure S13b). A median filter with a kernel size of 3 and a morphological transformation by kernel size of 4 was considered a noise removal algorithm. The median filter works well at removing the noise of the 3rd drop, but it was not able to follow the original diagram in some areas and some

displacement occurs (Figure S13b, yellow asterisks). The morphological transformation works based on erosion followed by dilation. This algorithm removed the noise of the 3rd drop and followed the original diagram better than the median filter. The noise removal algorithms could not remove the noise of the 2nd drop.

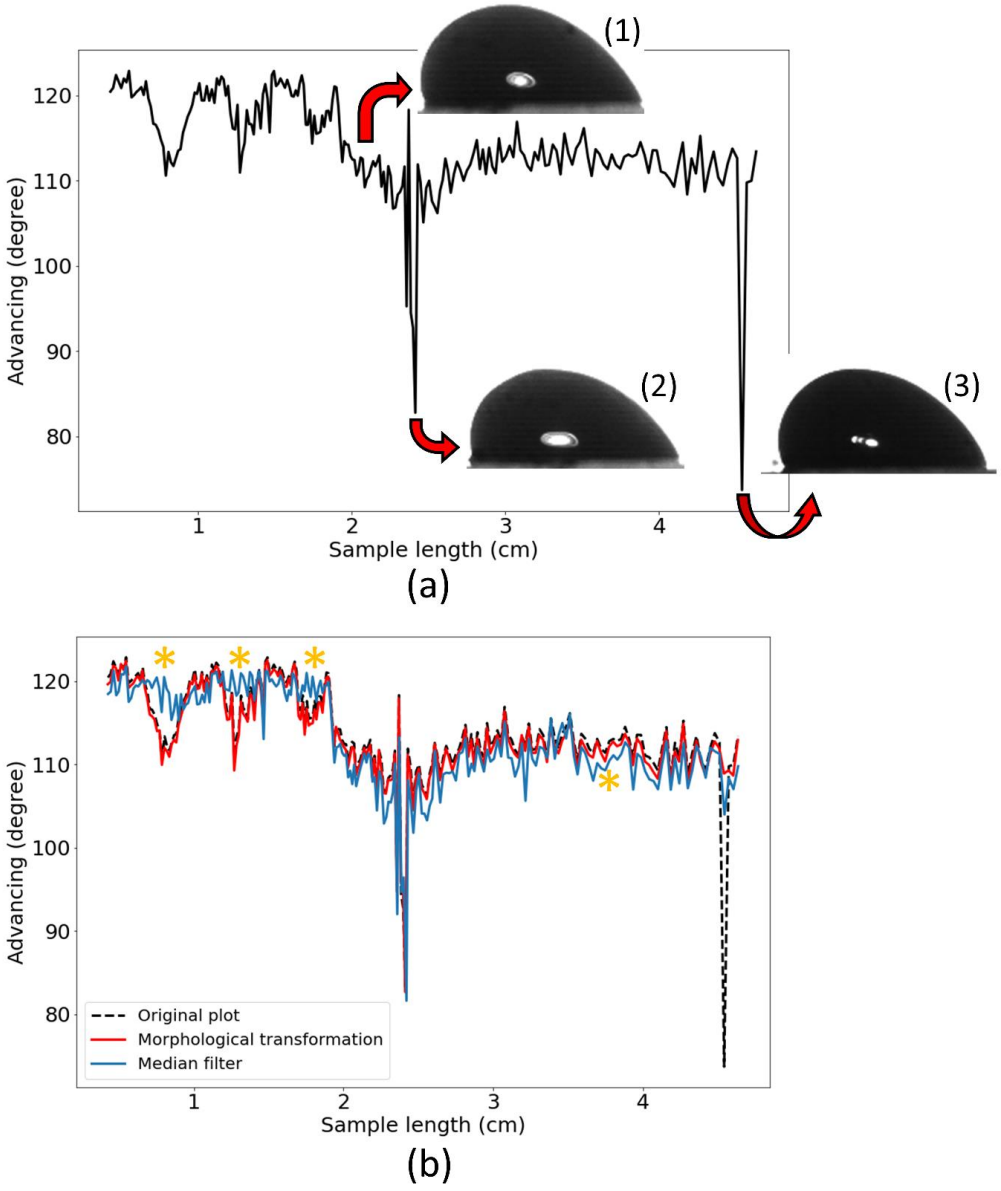


Figure S13. The effect of noise removal algorithms on the advancing diagram. a) Advancing diagram related to a sliding drop video, a_1: A normal drop image without any noise, a_2: A drop image with a noise close to the drop edge, a_3: A drop with a detectable noise. b) The black dotted plot is the original plot before using noise removal algorithms. The blue line is the advancing diagram after using the median filter method with a kernel size of 3. The red line is

S16

the advancing diagram after using the morphological transformation method with a kernel size of 4. Both the median filter and morphological transformation are able to remove the 3rd noise. But the median filter may change the drop edge and CA diagram in some cases. The yellow asterisks represent displacement caused by the median filter.

The proposed drop video analysis toolkit uses the morphological transformation method. However, morphological transformation does not come without side effects. The receding part of the drop is usually the most sensitive to this method. Erosion may detect the very end of the receding part as noise when receding is very low. It happens when the kernel size is large (Figure S14a). The larger the kernel size, the greater the displacement. However, the smaller kernel sizes can eliminate most of the noises and keep the receding value close to the original value (Figure S14b). In the proposed program default kernel size is considered 3.

S17

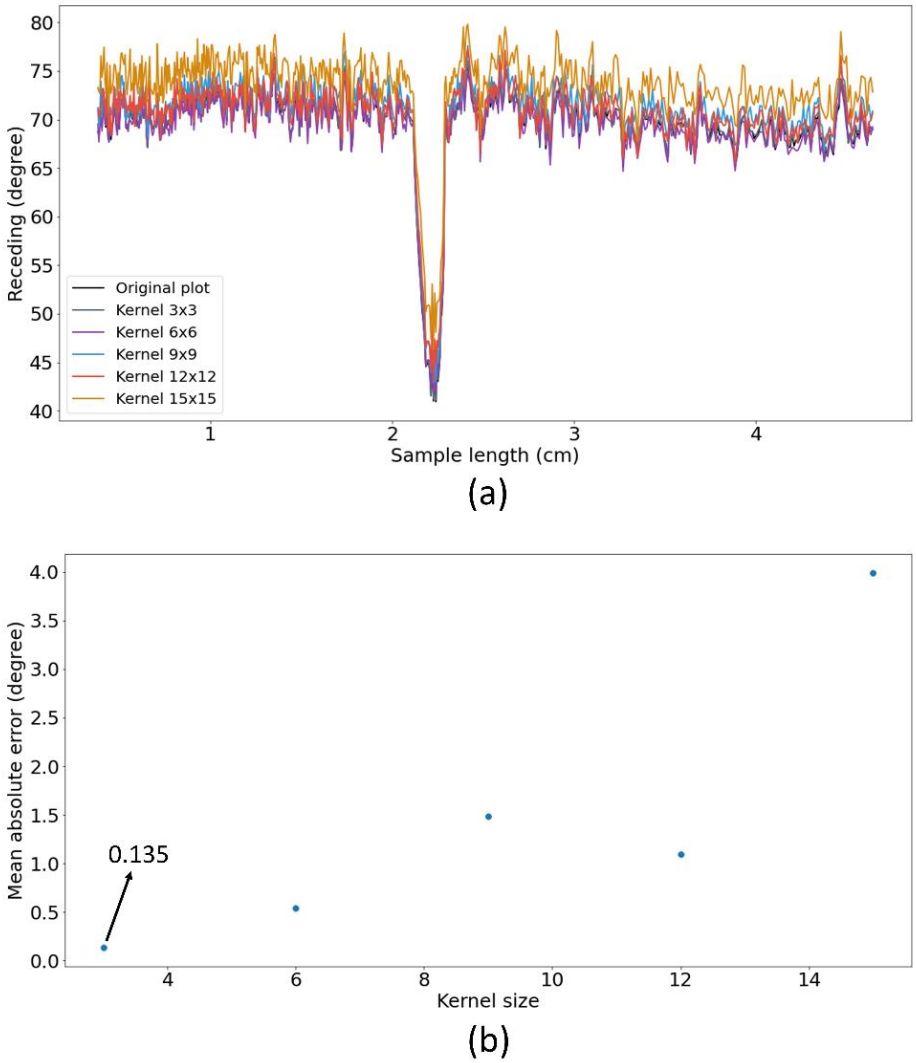


Figure S14. The effect of morphological transformation kernel size on the receding diagram. a) Visualizing how different morphological transformation kernel sizes affect the receding angle. The larger the kernel size, the greater the displacement. b) The morphological transformation effect on receding based on mean absolute error and kernel size. Kernel size 3 results in an MAE of only 0.135, which is very small and efficient.

S18

Baseline and tilt angle displacement

Defining the exact value of baseline and tilt angle is crucial to analyzing video correctly. The sensitivity of the drop profiles such as velocity, drop length, advancing angle, and receding angle to these two parameters are studied (Figure S15). Low-value displacements of baseline and tilt angle do not affect velocity and drop length. If the error in determining the baseline and tilt angle increases, the velocity and the drop length may also be affected. In the context of advancing and receding angles, existing error in baseline and tilt angle values is influential. Displacement in baseline, 2 rows, can shift the whole of the advancing angle and receding angle diagrams up to 5° values up or down. Also when 0.2° displacement happens in tilt angle, advancing angle and receding angle diagrams in some parts may be affected up to 5°.

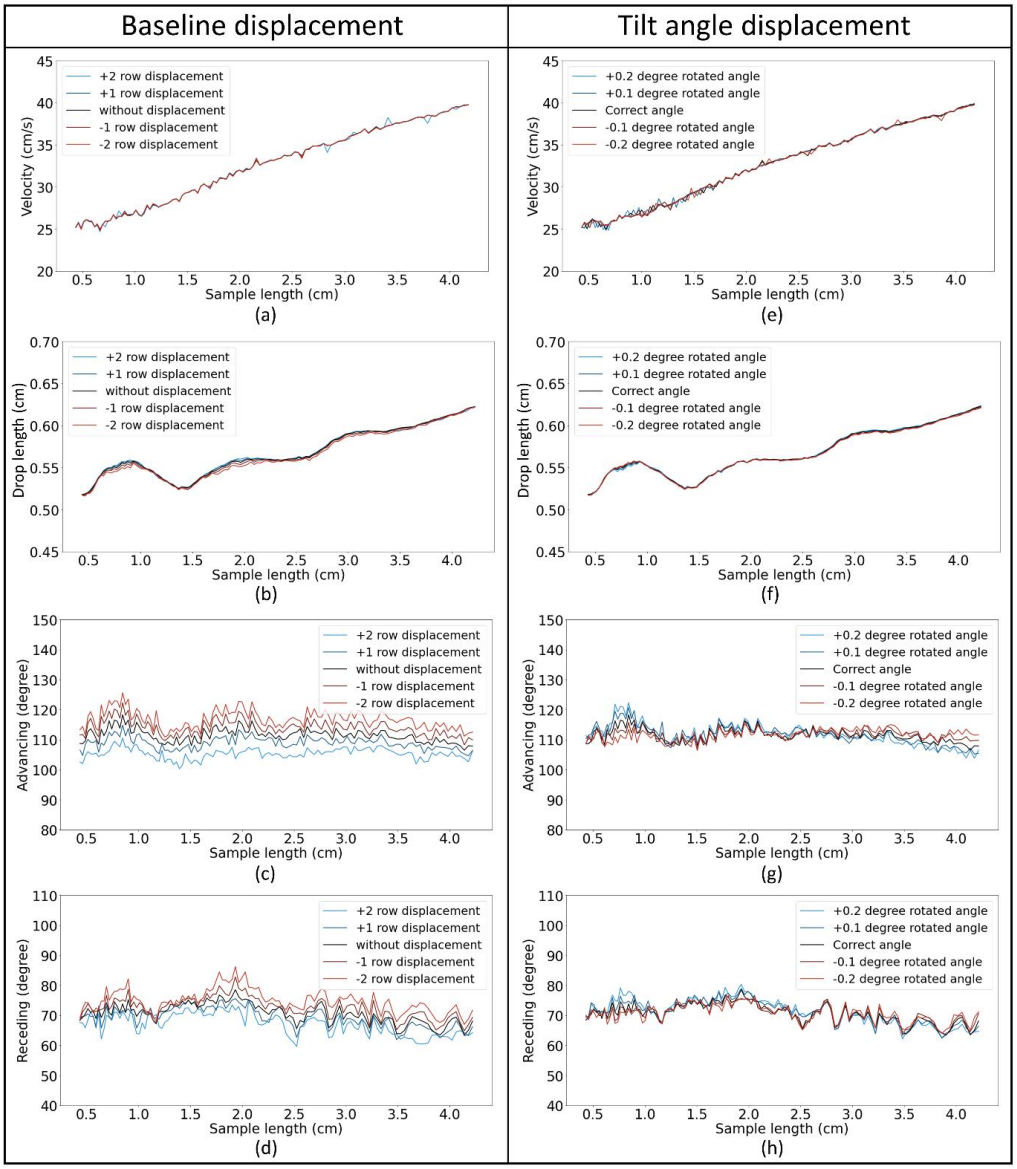


Figure S15 A visualization of baseline and tilt angle displacement and their effect on drop profile. The effect of baseline displacement on velocity (a), drop length (b), advancing angle (c), and receding angle (d). The effect of tilt angle displacement on velocity (e), drop length (f), advancing angle (g), and receding angle (h)

S20

Baseline detection

A parameter that is important for extracting the CAs is the position of the baseline. Existing methods for determining baseline location are based on edge detection algorithms [1, 2]. In most cases, the receding side of the drop had not a sharp reflection. Therefore, the baseline location at the receding side is not clear. To locate the baseline, we used only the advancing side. When the surface is transparent, edge detection algorithms have difficulty detecting the transition line between the real drop and its reflection (Figure S16). It happens because the reflection of the drops on the transparent surface does not have enough contrast. Consequently, the shape of the reflection does not exactly match that of the drop, and using edge detection methods can be misleading.

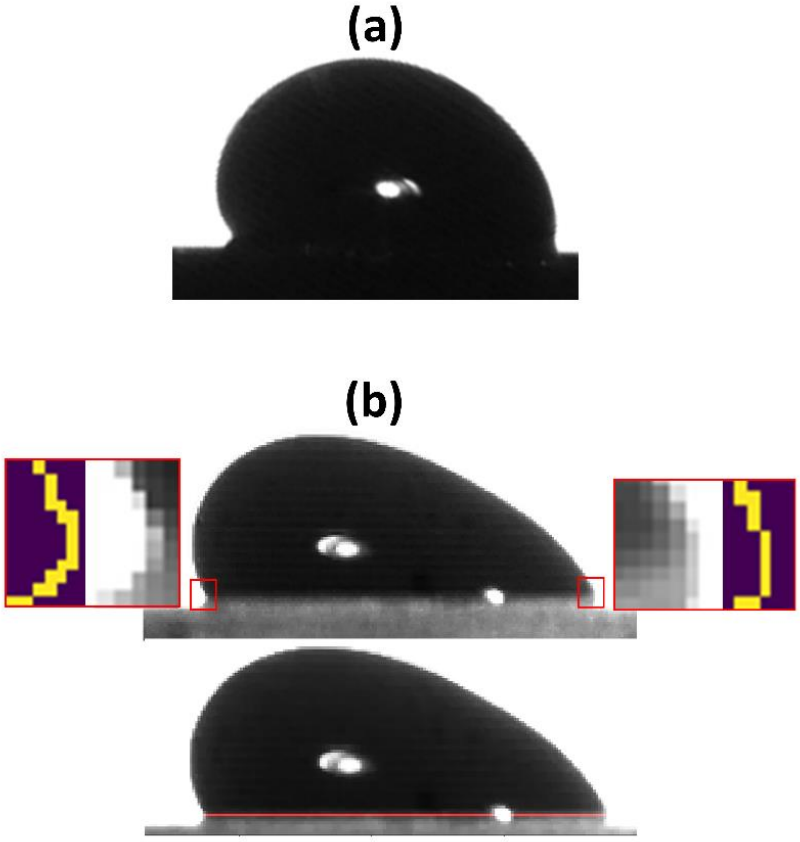


Figure S16. Baseline detection of drops on transparent and non-transparent samples: a) A non-transparent sample with a clear baseline. With this kind of sample, edge detection algorithms can easily find the baseline. b) The drop baseline on transparent samples is not distinct enough to be detected by edge detection algorithms. This image shows the detected edge using the canny method. No specific pixel represents the baseline in detected edges. The color-based proposed method can be applied in this case. The detected baseline (red line) is the output of the proposed method.

S22

Here, we employ a new approach to detect the baseline on transparent surfaces, which is not based on edge detection algorithms. We analyze the intensity of pixels in our images, which is a value between 0 and 255 (0 represents black and 255 represents white). We exemplarily explain our procedure with an image showing the bottom of a drop and parts of the sample surface (Figure S17a). In this image, the drop appears white and thus has composed of pixels with intensity values close to 255. We calculate the average intensity values inside each row of the matrix (Figure S17b). Rows 1 to 20 show a slight increase in intensity due to an increase in drop size. From rows 20 to 35, there is a significant decrease in intensity and then the intensity plateaus. The latter indicates the mirror image of the drop exhibiting an almost constant but lower average intensity. The baseline is in the transition and in order to define its position we calculated the intensity derivative (Figure S17c). We attribute the minimum value of this plot to the baseline of the drop. In order to verify our procedure, we calculated the baseline separately for every frame of a video that contains 175 frames. On average, the standard error for baseline detection was 0.44 pixels, i.e. less than one pixel. Thus, the approach seems robust and from each video we can extract a reliable and constant baseline.

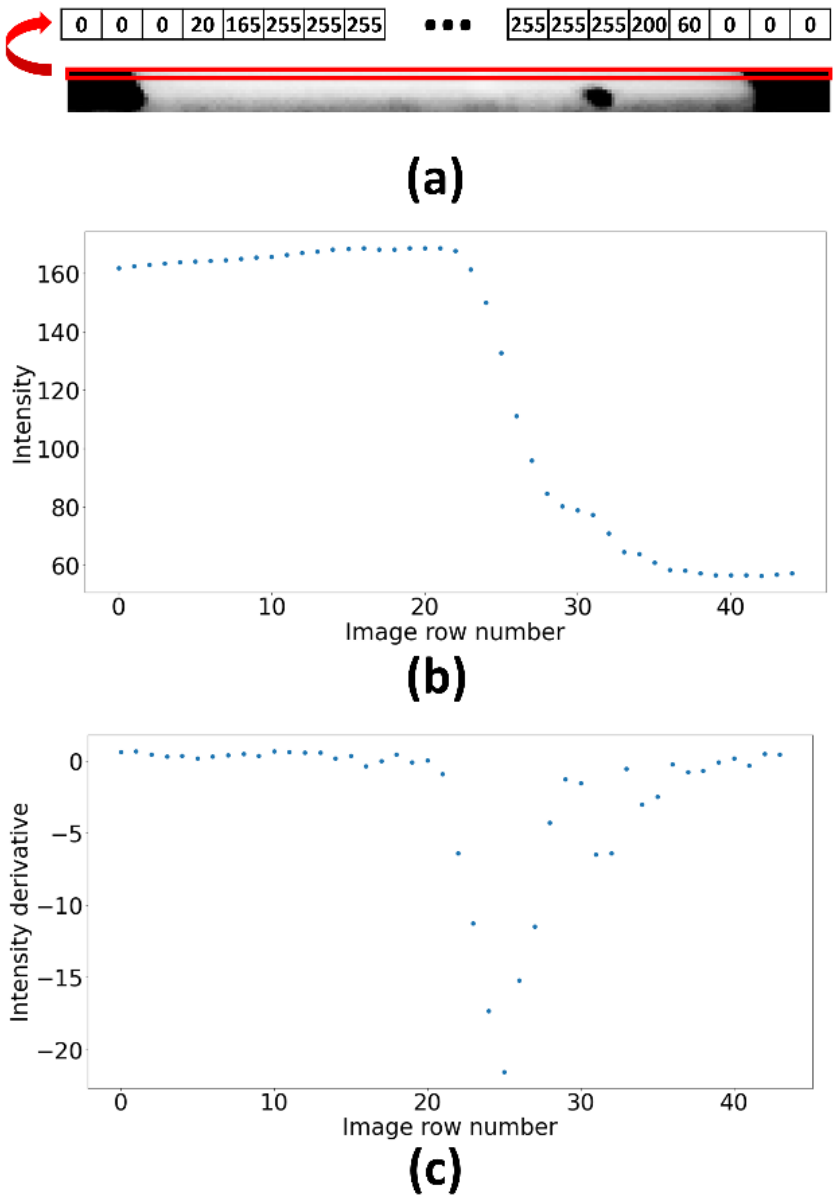


Figure S17. a) A representation of the first row of the numerical matrix of the drop image b) Average pixels intensity for each image row is decreasing in a specific area due to entering from the drop part to the reflection part. c) Derivative of Average pixels intensity for each image row with a minimum peak in the baseline position due to changing the environment.

S24

Measuring sphere diameter using super-resolution

We used images of a 5 mm diameter sphere that has a diameter similar to a drop with volume of 35 µl (Figure S18a). A standard deviation of +/- 1.6 µm was obtained after repeating the measurement ten times. The sphere was placed in the middle of the sliding drop setup and an image was taken (Figure S18b). After that, we treated the image with the Bicubic and our modified ESPCN procedure. For both procedures, we calculated the edge using a canny edge detection filter (Figure S18c). Then, we kept only the outer pixels of the sphere's edge horizontally, which is why the upside of the sphere is not connected. For further analysis, we calculated 20 times the diameter for the Bicubic and modified ESPCN model (Figure S18c). The average diameter for the Bicubic evaluation method was 5050 +/- 16.6 µm. The average diameter for the modified ESPCN evaluation was 5015 +/- 6.7 µm. The modified ESPCN has a less standard deviation.

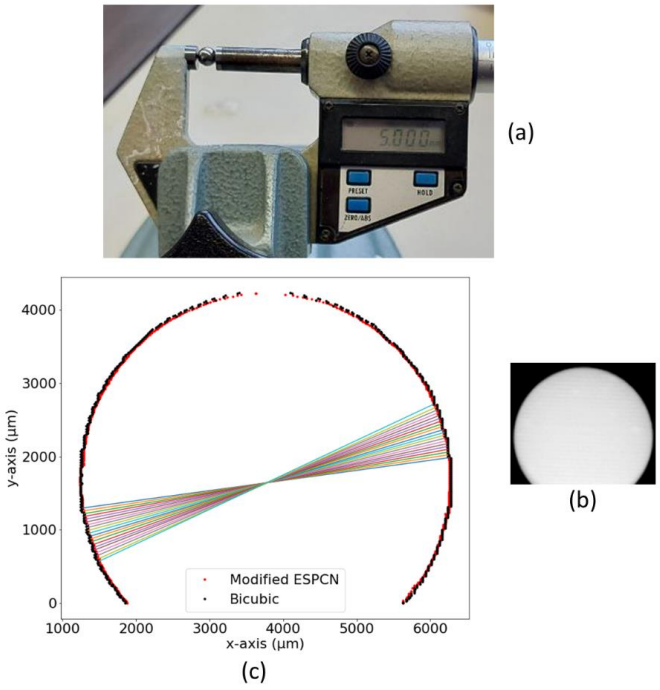


Figure S18. Sphere diameter measurement using Bicubic and super-resolution methods. a) Measuring real diameter of the sphere b) Sphere image in sliding drop setup c) Measuring sphere diameter 20 times using Bicubic and super-resolution methods.

Supporting video description

The video shows a sliding drop on a sample with a defect in the middle. The first drop in the upper part of the video is a real drop image after preprocessing steps. The main steps of preprocessing included calculating the tilt angle and making the frames horizontal, removing noises and background, and detecting drop position (red lines). The second drop image which is bigger than the first one is the drop image after using a super-resolution model. Below that, the drop contour is extracted and different parameters including CAs, drop height, and drop length are displayed. On the left, four figures are getting plotted to analyze how a drop slides on a sample with a defect.

S26

Corresponding Author

Rüdiger Berger, Max-Planck-Institut for Polymer research (MPI-P), Ackermannweg 10, 55128 Mainz, Germany. Email: berger@mpip-mainz.mpg.de

References

1. Kalantarian, A., R. David, and A.W. Neumann, *Methodology for High Accuracy Contact Angle Measurement.* Langmuir, 2009. **25**(24): p. 14146-14154.

2. Atefi, E., J.A. Mann Jr, and H. Tavana, *A robust polynomial fitting approach for contact angle measurements.* Langmuir, 2013. **29**(19): p. 5677-5688.

Chapter 4

Publication 2

4.1 Estimating sliding drop width via side-view features using recurrent neural networks

4.1.1 Summary and author contribution

This publication builds on the side-view time series derived in P1 and casts the estimation of the drop's front-view width as an extrinsic time-series regression task. Rather than using a single empirical correlate such as the center velocity, it investigates multivariate relationships between multiple side-view features and the unknown width, benchmarking a broad range of regression and sequence-learning approaches. The comparative evaluation highlights the advantages of recurrent architectures for this setting. Overall, P2 introduces a machine-learning solution for continuous width estimation along the full sliding trajectory without requiring additional experimental hardware.

The author played a leading role in defining the research question, designing the modeling approach, and implementing the machine-learning framework. The author developed the data processing and model training routines, conducted the comparative analyses, and interpreted the results in the context of wetting and sliding dynamics. The author wrote substantial parts of the manuscript, including the sections on methodology, results, and discussion.

4.1.2 Scientific publication

www.nature.com/scientificreports

scientific reports

Check for updates

OPEN

Estimating sliding drop width via side-view features using recurrent neural networks

Sajjad Shumaly[1], Fahimeh Darvish[1], Xiaomei Li[1], Oleksandra Kukharenko[1], Werner Steffen[1], Yanhui Guo[2], Hans-Jürgen Butt[1] & Rüdiger Berger[1]✉

High speed side-view videos of sliding drops enable researchers to investigate drop dynamics and surface properties. However, understanding the physics of sliding requires knowledge of the drop width. A front-view perspective of the drop is necessary. In particular, the drop's width is a crucial parameter owing to its association with the friction force. Incorporating extra cameras or mirrors to monitor changes in the width of drops from a front-view perspective is cumbersome and limits the viewing area. This limitation impedes a comprehensive analysis of sliding drops, especially when they interact with surface defects. Our study explores the use of various regression and multivariate sequence analysis (MSA) models to estimate the drop width at a solid surface solely from side-view videos. This approach eliminates the need to incorporate additional equipment into the experimental setup. In addition, it ensures an unlimited viewing area of sliding drops. The Long Short Term Memory (LSTM) model with a 20 sliding window size has the best performance with the lowest root mean square error (RMSE) of 67 µm. Within the spectrum of drop widths in our dataset, ranging from 1.6 to 4.4 mm, this RMSE indicates that we can predict the width of sliding drops with an error of 2.4%. Furthermore, the applied LSTM model provides a drop width across the whole sliding length of 5 cm, previously unattainable.

Keywords Sliding drops, Drop width estimation, Multivariate sequence analysis, Recurrent neural network (RNN), Long short-term memory (LSTM), Gated recurrent unit (GRU), Bidirectional LSTM (BiLSTM), Convolutional neural network (CNN), Convolutional LSTM (ConvLSTM)

Researchers have employed side-view video recordings of sliding drops to analyze the physico-chemical behavior of the liquid to solid interface surface[1,2]. Investigation of the wealth of phenomena of sliding drops requires a measurement of the drop shape from the front-view[3–5]. Often extra cameras or mirrors are added for front-view analysis. This approach, while common, brings practical challenges. In continuing, we will present two examples highlighting the significance of drop width and the need for front-view observation.

On homogeneous surfaces, hydrodynamic dissipation increases with velocity. Recently, Li et al. studied drops sliding down an inclined surface[3] and reported an empirical equation that describes the velocity-dependent friction force:

$$F_f = F_0 + \beta w U \eta \tag{1}$$

where, F_0 is the friction force extrapolated to velocity $U = 0$, β is a dimensionless friction coefficient, w is the width of the drop while sliding, and η is the viscosity of the liquid. The geometry of drops and their kinetic contact angles sliding down an inclined plane change with velocity. These parameters can be readily measured in a side-view (Fig. 1). However, within a standard sliding drop experiment, determining the drop width while sliding remains a challenging task.

In addition to homogeneous surfaces, topographical and chemical variations spanning from nano- to micrometers on surfaces lead to the pinning of the three-phase contact line[6]. Pinning of the contact line increases contact angle hysteresis (CAH), and drop friction, thus decreases drop velocity. The interactions between solid surfaces and liquids are described by the liquid–air surface tension (γ), the width of the contact area (w), and the apparent rear and front contact angles of the drop (θ_a, and θ_r)[4,5]:

[1]Max Planck Institute for Polymer Research (MPI-P), Ackermannweg 10, 55128 Mainz, Germany. [2]Department of Computer Science, University of Illinois Springfield, Springfield, IL, USA. ✉email: berger@mpip-mainz.mpg.de

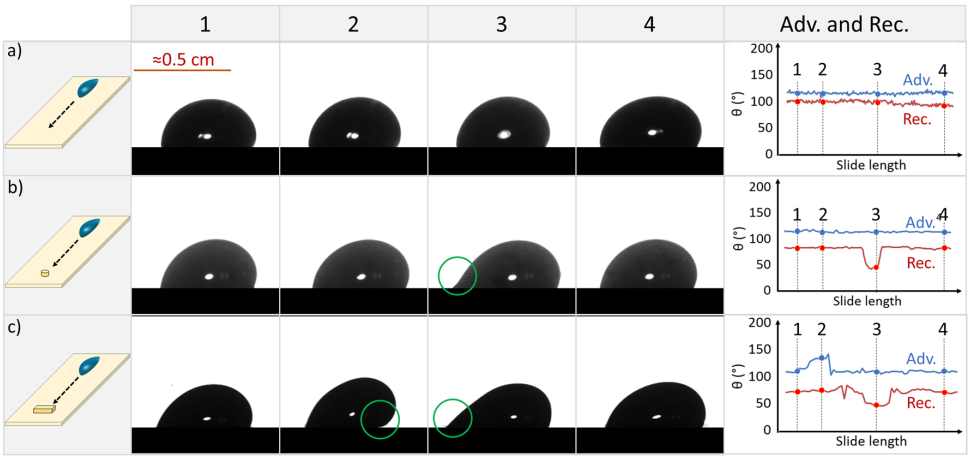


Figure 1. Side-view images of 32 µL water drops at different positions along their path. (**a**) The Thiols_Au sample has no large defects or heterogeneity. The tilt angle was 30°. (**b**) The PFOTS_Si sample has a single cylindrical defect (D-Cy-800), with a height of 31 µm, which affects the dynamic receding contact angle (θ_r) of the drop (green circle). The dynamic advancing contact angle (θ_a) is not changed significantly. The tilt angle was 35°. (**c**) The PFOTS_Si sample has a block defect (D-Bl-3000), with a height of 174 µm, that affects both dynamic advancing and dynamic receding contact angles considerably due to its larger size (two green circles). The tilt angle was 50°. By increasing the defect size, the tilt angle is increased to allow the drop to pass the defect. In the specific examples presented above, we only plotted θ_a and θ_r, but additional parameters of the drop such as drop length, velocity, drop height, and middle line angle changes as well.

$$F_{LA} = k\gamma w(cos\theta_r - cos\theta_a) \tag{2}$$

where F_{LA} is the lateral adhesion force, θ_a is the advancing angle, θ_r is the receding angle, and k is the geometry factor[7]. The lateral adhesion force has been related to external forces that cause a drop to slide, such as gravitational[8], drag by a micropipette[9], and centrifugal[10] forces. Equation (2) is often called the Furmidge equation and was first reported in the 40ies[11–14]. The Furmidge equation indicates that the drop width is directly related to lateral adhesion force. The Furmidge equation was also taken to calculate F_{LA} for sliding drops. Interestingly, the calculated F_{LA} is consistent with forces calculated by the equation of motion[3] assuming $k = 1$. Thus, measuring θ_a, θ_r and w will allow us to calculate F_{LA}.

However, determining the drop width of sliding drops is challenging. Drop width data can be collected by recording bottom-view or front-view videos of sliding drops. Bottom-view imaging is restricted to transparent substrates. Front-view imaging of drops over a sliding length of ≈1.5 cm is feasible by installing a second, time-synchronized high-speed camera[15]. Also, a second high-speed camera can be omitted by installing two mirrors at the back and front of the drop sliding length[16]. One mirror reflects light from the light source and via the other front-view videos are recorded. The primary challenge in the experiment lies in optimizing lighting conditions for the additional mirrors/second camera. Positioning mirrors in the optical path is tricky, and can lead to a reduced contrast in the final front-view images due to light reflection. Introducing another camera necessitates an extra front-view illuminator, doubling the equipment on a platform that must rotate 90°. More importantly, in both cases, the direction of drop motion towards the camera or mirrors limits the focus area for front-view images to approximately 1.5 cm, significantly narrowing the observable area.

The question we faced was whether we could estimate drop width based solely on side-view measurements. The sliding behavior of a drop involves changes in the dynamic advancing (θ_a) and receding (θ_r) angles, drop length, velocity, drop height, and middle line angle. These factors can interact in complex ways, creating intricate patterns. For drops sliding on homogeneous surfaces, θ_a and θ_r change monotonically throughout the entire sliding path, as shown exemplarily for a perfluorodecanethiol monolayer on gold coated glass (Thiols_Au) sample (Fig. 1a). In cases where an obstacle is encountered along the drop's route, alterations to the dynamic contact angles occur. For a cylindrical obstacle with a diameter of 800 µm (D-Cy-800), we measured that θ_r changes while only minimally influencing the θ_a on a 1H,1H,2H,2H-perfluoroctyltrichlorosilane coated silicon wafer sample (PFOTS_Si, Fig. 1b). For a bigger block defect, 800 µm thick and 3000 µm length (D-Bl-3000), on a PFOTS_Si surface, both the θ_a and θ_r change considerably (Fig. 1c).

Machine learning models were successfully deployed in surface science. In a study on water drop impact on supercooled surfaces, two models were developed to predict the spreading dynamics of the drop upon impact and classify the resulting icing patterns[17]. These models aimed to forecast the degree of surface supercooling corresponding to the observed icing patterns. DeepAngle is a machine learning-based approach, to accurately determine contact angles in tomography images of porous materials[18]. Traditional methods for measuring 3D angles face challenges in voxelized spaces. Therefore, deep learning models were applied to estimate interfacial

angles directly from images, instead of computationally intensive grid-based approaches. The latter enhances accuracy to 16% while reducing computational costs 20-fold. Tanaka et al., developed a neural network model, which allows prediction of contact angles for a wide range of metals and oxide surfaces[19]. A CNN-based method was applied for contact angle measurement of the moving drops to overcome traditional algorithm limitations due to optical distortions by changes in the focal length[20]. The proposed method exhibits robustness against higher Gaussian Blurring values. Recently, we presented a CNN-based super-resolution technique that achieved a more precise analysis of sliding drops[21]. The reported approach led to a 21% increase in accuracy for contact angles below 90° and a 33% improvement for contact angles above 90°.

Here, we explore different machine learning models for estimating drop width based on drop parameters recorded dynamically from the side-view. Our contributions are outlined as follows:

1. We introduce a machine learning-based approach. This approach does not require additional mirrors or cameras traditionally used for a frontal view of the sliding drop. The machine learning-based approach significantly simplifies the experimental setup.
2. For the first time, our machine learning-based approach enables the continuous monitoring of a sliding drop along its entire path, which typically is 5 cm. The viewing area is not limited by the type of optics. This capability overcomes limitations that restrict observations to the final centimeter of a sliding drop.
3. We conducted a comparative analysis between regression models and multivariate sequence analysis (MSA) models. This comparison delves into the independence or interdependence of drop parameters relative to its preceding and subsequent values. With this comparison, researchers are able to select the most appropriate model to meet their specific needs.

In the "Materials and Methods" section, we delve into the cutting-edge methods currently used for sequence analysis. We describe in detail the experimental procedures employed to collect the dataset, and prepare the relevant samples. Then we describe the dataset's structure and explain our training process. In the "Results and Discussion" section, we evaluate the precision of various regression and MSA (Multivariate Sequence Analysis) algorithms in predicting drop width. We further authenticate the performance of our optimal model using a sample with different defect height. Additionally, we perform sensitivity analysis to equip researchers with the necessary insights to effectively apply our developments in their investigations. The developed LSTM model and its updates is accessible in the GitHub repository[22].

Materials and methods

Sequence analysis methods

Many real-world prediction problems have been successfully addressed using deep learning methods, including time-series forecasting[23,24]. Recurrent neural network (RNN) is a deep learning time series analysis method that has been gaining significant interest recently[25–27]. RNNs enhance the accuracy of predictions by using their recurrent architecture, which allows them to capture and utilize sequential information and temporal dependencies in data. The effectiveness of RNNs stems from their capacity to incorporate past events and their utilization of shared parameters or weights. Each RNN cell takes an input and combines it with the previous hidden state to produce an output and a new hidden state.

However, deep neural networks with a fully connected architecture are prone to the problem of vanishing gradient[28]. A vanishing gradient occurs when the network is unable to update its weights due to its activation functions and network architectures, which cause gradient values to be too small during backpropagation[29]. RNN is a biased model and gives high importance to recent occurrences, reducing its effectiveness[30]. Thus in applications, RNN is refined into RNN-based models, such as Long Short-Term Memory (LSTM) and Gated Recurrent Units (GRU) models.

A solution for the vanishing gradient problem is a method called LSTM[31,32]. The fundamental concept underpinning the LSTM architecture involves a memory cell with the ability to sustain its state throughout time. Coupled with nonlinear gating units, these components oversee the inflow and outflow of information from the cell[33]. Each LSTM cell has three gates: a forget gate, an update gate, and an output gate. The LSTM network architecture can learn from past data using its gates in order to remember the past data and thus creates a prospective model based on the past and current data. In this way, LSTM models can capture sequence pattern information more efficiently[31]. In LSTM, the forget gate's role is to determine which data from the previous cell state should be eliminated for the current time step. While, the update gate is responsible for selecting what new information is eligible for storage in the cell state, and the output gate manages the information available for output based on the cell state.

The Gated Recurrent Unit (GRU) emerges as an innovative approach aiming to streamline the complexity inherent in LSTM[34]. GRU, like LSTM, utilizes gating techniques to regulate the network's information flow selectively. GRU only uses the reset gate and the update gate, as opposed to LSTM's three gates. The update gate regulates the combination of incoming input with the old state, whereas the reset gate regulates the extent to which the past information is ignored. GRU is more computationally efficient owing to this streamlined architecture, which has fewer parameters than LSTM[35]. Supporting information (SI-RNNs' architectures) includes the RNN, LSTM, and GRU architectures as well as their formulas.

The Bidirectional LSTM (BiLSTM) model is an advancement of LSTM and involves two LSTM cells, namely the forward and backward LSTMs[36]. In BiLSTM, the input sequence is processed twice, first from left to right and then from right to left. The LSTM takes into account all previous events, whereas the BiLSTM considers both past and future events. BiLSTMs are thus superior to LSTMs in some cases[37].

The field of computer vision has been revolutionized by convolutional neural networks (CNNs)[38]. CNN was successfully applied in various fields and researchers used it for time series analysis[39]. In general, CNNs can handle spatial auto-correlated data, detecting patterns in short-term, and time series data with local dependencies. However, they are not typically trained to handle long temporal relationships[40]. Therefore, a time-series model which exploits the benefits of both CNNs and LSTMs, such as Convolutional LSTM (ConvLSTM), will be able to capture also long-term dependencies [41–43].

A number of models have been proposed to tackle the challenge of interpreting deep models and identifying important features [44–46]. Gradient-weighted Class Activation Mapping (Grad-CAM) is an interpretation model that was originally designed for image processing[47]. Grad-CAM is versatile and is extended for text and sound analysis[48–50]. Similar to Grad-CAM, which calculates the gradients of a target output with respect to convolutional layer activations to identify important regions in an image, we used gradient-based feature importance to assess the influence of input features on the output of the LSTM model.

Data gathering

Drops of distilled water (< 1 μS cm^{-1}; Gibco, Thermo Fisher Scientific) with a volume of 32 μl were placed on top of a tilted plane using a peristaltic pump (MINIPULS 3, Gilson) connected to a grounded, blunt syringe needle (1.5 mm outer diameter) (Fig. 2a). The liquids were dropped from a height of approximately 5 mm, which enabled them to detach from the syringe before reaching the surface. The optical breadboard with mounted components can be rotated from 0 to 90°. By rotating the entire setup the alignment of the optical setup is kept[16]. In particular, the angle between the video camera and the sample stays constant close to 0°. Videos of sliding drops were recorded with a high-speed camera (FASTCAM Mini UX100, Photron) equipped with a TitanTL telecentric lens (×0.268, one inch, C-mount, Edmund Optics). The front-view of the sliding drops was captured at the same time by reflecting the backlight from the telecentric backlight illuminator (Edmund Optics) using two parallel mirrors ($25 \times 36\,mm^2$ protected silver mirror; PFR10-P01, Thorlabs) on both sides of the sample. The temperature and humidity during the experiment were approximately 20 ± 1 °C and 15–30%, respectively. A standard backlight illuminator was used limiting frame rates to 500 fps. At higher frame rates the contrast of images reduces and the drop shape cannot be analyzed.

We employed the 4-segment super-resolution optimized-fitting (4S-SROF)[21,51] toolkit to extract drop velocity, drop center height, drop length, θ_a, θ_r and the drop's middle line angle from the recorded videos (Fig. 2a). The velocity is calculated based on the position of the center of the drop (X of the center in the figure). All experiments were repeated several times at different tilt angles and analyzed by the 4S-SROF toolkit, resulting in our dataset (SI-Data distribution). It was essential to gather data during changes in the sliding drop pattern. For instance, when changing the surface chemistry, the drop velocity was altered. Surface defects were introduced, to increase the model's generality. Also, in produced samples the height of the defects changes, and we varied the viscosity of the liquid by using mixtures of water and glycerol of varying concentrations. We used DI water with a viscosity of 0.92 mPa s, 20% glycerol-water mixture (1.7 mPa s), 30% glycerol-water mixture (2.5 mPa s), and 40% glycerol-water mixture (3.8 mPa s).

Sample preparation

Samples with block and cylindrical defects were produced on silicon wafers. First, wafers were cleaned by ultrasonication in acetone twice, followed by 2-propanol to remove organic impurities. Next, cleaned Si-wafers were exposed to an O_2 plasma (140 W, 5 min, 0.3 mbar, Femto–Diener GmbH, Germany). A filtered nitrogen gun that blows away microscopic fibers and dust was used to dry and remove the small particles from the Si-wafers. Then, 1 ml SU-8 (either GM1060, or GM1070, Gersteltec Sàrl, Switzerland) was dropped on Si-wafers. In the GM1060 case, the rotation speed was ramped up from 500 rpm (7 s) to 1000 rpm (40 s) for the validation block defect. In the GM1070 case, the rotation speed was ramped up from 500 rpm (7 s) to 700 rpm (40 s), from 500 rpm (7 s) to 1000 rpm (40 s), and from 500 rpm (7 s) to 1500 rpm (40 s) for different block defects. Also, in the GM1060 case, the rotation speed was ramped up from 500 rpm (7 s) to 850 rpm (40 s), and from 500 rpm (7 s) to 600 rpm (40 s) for different cylindrical defects. To evaporate solvents from the thin layers of resist, a soft baking process was carried out on a hotplate (65 °C for 30 min, 95 °C for 2–4 min, and 65 °C for 30 min). After gradually cooling down to room temperature, the samples were mounted to a mask aligner (MJB3, Süss Microtec, Germany). UV light through a photomask exposed for 8 s with 290 J/cm^2 energy. After that, post-exposure baking for 1–2 min at 95 °C was performed. Unpolymerized SU-8 was removed by rinsing with 1-methoxy-2-propanol acetate (CAS# 108-65-6). Finally, samples are washed for 1 min in 2-propanol. Before fluorination, samples were activated by the oxygen plasma for 5 min 140 W, 0.3 mbar. 0.5 mL of 1H,1H,2H,2H-perfluoroctyl trichlorosilane (PFOTS, 97%, CAS:78,560-45-9, PFOTS) was selected due to its low boiling point of 180 °C, which was beneficial for easy evaporation, and its high reactivity for deposition. Fluorination was performed in a CVD chamber under 40–50 mbar pressure at room temperature for 30 min. To remove uncrosslinked PFOTS, the samples were kept at 0 mbar for 30 min and then washed and rinsed with ethanol and Milli Q water. Also, defect-free PFOTS_Si samples were produced using the given procedure.

To prepare samples with gold-coated glass substrates (gold substrate), 5 nm Chromium and 35 nm gold were sputter coated onto glass substrates subsequently (BalTec MED 020). The gold substrates were used immediately and directly without further cleaning. Thiols-gold samples were prepared by submerging gold substrates in a 1 mM ethanolic 1H,1H,2H,2H-perfluorodecanethiol (≥ 96.0%; Sigma-Aldrich) solution for 24 h. Before using, we rinsed the samples with absolute ethanol to remove unbounded thiols and dried them with air blowing. To get Teflon-gold samples, ~ 60 nm Teflon film was coated on the gold substrate by dip coating with a pulling speed of 10 mm/min from a solution of 1wt% Teflon AF 1600 (ε = 1.9; Sigma-Aldrich) in FC-75 (97%, Fisher Scientific). Finally, the Teflon samples were annealed in a vacuum oven at 160 °C for 24 h before use.

www.nature.com/scientificreports/

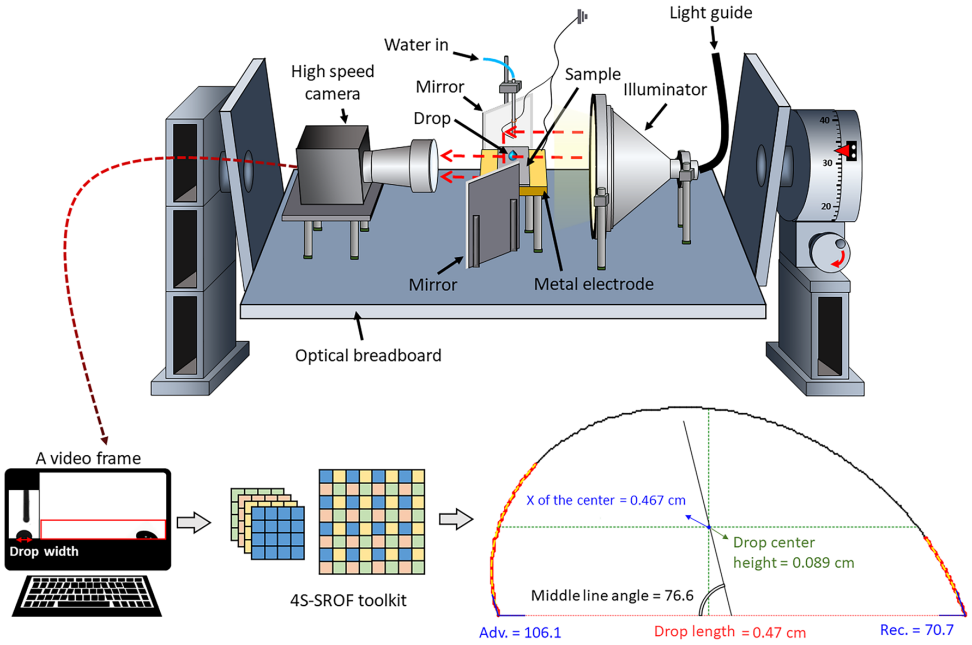


Figure 2. (**a**) A sketch of the tilted plane experimental setup. Traces of sliding drops were recorded by a high-speed camera equipped with a telecentric objective. Videos were stored on a computer. Drop contact angles, drop length, drop center height, median line angles, and velocity were extracted by a 4S-SROF toolkit. (**b**) Snapshot of sliding drop in side- and front-view. (**I**) One frame from a video which was recorded without mirrors. Here the slide length for the drop is ≈5 cm. (**II**) The mirrors that are required to image a drop in front-view occupy ≈1.5 cm of the camera field of view. Furthermore, the initial ≈2 cm of the video will be out of focus in the front view, as demonstrated exemplarily by magnifying the interface between the liquid and the air. The use of the mirror reduces the effective area for recording the sliding length of the drop from ≈5 to ≈1.5 cm.

Data structure and training process

Without the use of mirrors in our tilted plane setup, the field of view covers a sliding length of 5 cm (Fig. 2b_I). The installation of mirrors cut the field of view of the camera; The front-view image with a field of view of ≈1.5 cm and the side-view with a length of ≈3.5 cm (Fig. 2b_II). Please note, when the drops are situated in the initial 2 cm of the sample, they are out of focus of the telecentric objective due to the large distance from the reflecting mirror. Only in the last ≈1.5 cm of the slide path drops are in focus.

The dataset consists of side-view images with their corresponding front-view images. The initial 2 cm of sliding length was excluded from the dataset, since precise drop width measurement was not possible. Thus, the initial 2 cm sliding length was not used for training models. Defects were fabricated on the last cm of samples, where we capture the drop shape in the side- and front-view.

The dataset was filtered to include only videos with more than 20 and less than 250 frames, aiming to ensure data consistency and relevance. Min–max scaling was utilized to standardize every aspect of the dataset within a range spanning from negative one to positive one. This process ensures that all features are on a similar scale, preventing any feature from dominating the analysis due to its magnitude. The data was segmented into smaller windows of 5, 10, 15, and 20 consecutive frames, with overlap, using a sliding window approach. This segmentation facilitates the analysis of sequential data and allows the model to capture temporal patterns effectively.

Our dataset consisted of 235 videos, including 13,301 frames. Each video contains different number of frames. The dataset was divided into two distinct subsets. A testing subset was created comprising 10% of the dataset (hold-out testing dataset), while the remaining 90% was allocated for training, in all cases.

We employed a grid search algorithm[52] to fine-tune the critical hyperparameters of each regressor, aiming to identify the optimal configuration as follow:

- **Multilayer perceptron:**
  - Number of layers: Varies from 1 to 4 with intervals of 1.
  - Number of nodes: Varies from 25 to 100 with intervals of 25.
  - Alpha values: Options include 0.01, and 0.1.
  - Learning rates: Options include 0.001, 0.01, and 0.1.
- **Gradient boosting:**
  - Number of estimators: Varies from 50 to 300 with intervals of 50.
  - Minimum samples split: Options include 2, 5.
  - Maximum depth: Options include none, 5, or 10.
  - Minimum samples leaf: Options include 1, 2.
  - Learning rates: Options include 0.01, 0.1, and 0.2.
- **Random forest:**

  Number of estimators: Varies from 50 to 300 with intervals of 50.
  Maximum features: Options include 'sqrt', 'log2'.
  Maximum depth: Options include none, 5, and 10.
  Minimum samples split: Options include 2, 5.
  Minimum samples leaf: Options include 1, 2.

In the case of MSA models, to prevent overfitting during training, a validation set was created by taking 20% of the training data. The validation set allowed monitoring of the model's performance on unseen data during the training process. This separation ensures that the model's performance can be evaluated on unseen data during the training process. The number of epochs for MSA models was set to 2500, as this was found to be the point at which the loss diagrams began to plateau. The observation of a plateau ensured that the model was able to converge to an optimal solution without overfitting the training data. Given the time-intensive nature of the training process, we bypassed the grid search algorithm for tuning MSA hyperparameters. Instead, we explored a single-layer LSTM configuration, experimenting with unit sizes of 32, 48, 64, and 128 to achieve a satisfactory structure. We employed a consistent structure across all models, involving a single layer with 48 units, whether it was LSTM, GRU, BiLSTM, or ConvLSTM. Additionally, a dropout rate of 0.5 and L2 kernel and recurrent regularization of 0.01 were consistently applied across all cases to prevent overfitting. The activation function employed was "tanh", and the optimizer used was "adam" for all MSA model configurations. This approach allowed for a balanced comparison of model performance under standardized conditions.

We utilized the mean square error (MSE, Eq. 3) as the loss function for MSA models. The MSE calculates the average of the squared differences between the predicted values and the actual target values. MSE is commonly employed for sequence analysis. In addition, we use root mean square error (RMSE, Eq. 4) and mean absolute error (MAE, Eq. 5) formulas to evaluate the accuracies:

$$MSE = \frac{\sum_{i=1}^{n}(x_i - \widehat{x_i})^2}{n} \tag{3}$$

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(x_i - \widehat{x_i})^2}{n}} \tag{4}$$

$$MAE = \frac{\sum_{i=1}^{n}|x_i - \widehat{x_i}|}{n} \tag{5}$$

where x_i is the true value, $\widehat{x_i}$ is the estimated value, and n is the total number of data points.

The choice of RMSE over MSE for representing the error results was motivated by RMSE's capability to retain the same unit as the original data. Utilizing RMSE enhances the ease of understanding and interpretation of the results. In this case, the error is calculated based on the unit of drop width, which is "µm". The utilization of two measures is justified by the fact that MAE assesses the overall error and provides insights into the general estimation quality of the developed model. On the other hand, RMSE penalizes errors by squaring them, thereby giving importance to large errors. The inclusion of both RMSE and MAE allows for a thorough evaluation of the model's performance, encompassing both the overall estimation accuracy and the consideration of large errors.

Results and discussion

Drop width estimation

In regression techniques, we assume that it is possible to estimate the drop width from each side-view image of a video (Fig. 3a_I). Hence, the drop width is an outcome of all features of the drop accessible from an individual side-view image. Regression techniques disregard the temporal dependencies that exist between time steps in the data. Therefore this approach is applicable for data sets where there is only a weak or negligible temporal dependency between the past and future data. To create a model that offers simplicity and interpretability, we used a linear regression model[53]. In addition to being useful for analyzing direct relationships between variables, linear regression can be utilized to assess the complexity of other models. To identify complex nonlinear correlations in the data, a multilayer perceptron model[54] was used. The multilayer perceptron model excels in managing intricate patterns and relationships. The choice of the random forest[55] and gradient boosting[56] algorithms stems from the aim to improve generalization and mitigate overfitting; their robustness and versatility are utilized to achieve the enhancement.

As an alternative, it is possible to estimate the width of the drop at a given slide length by considering the shape of the drop at the previous and following slide length, i.e. previous and following images of a video (Fig. 3a_II). Thus with MSA, temporal dependencies are taken into account. By considering the temporal dependencies between time steps, a superior outcome can potentially be achieved when addressing the problem using MSA instead of regression methods. This strategy is more intricate and resource-intensive. Moreover, these methods are not as easily interpretable as regression models. We will investigate for different scenarios whether the accuracy achieved through the MSA is significantly better than that achieved through a simple regression method. We utilized LSTM and GRU architectures because they are renowned for their ability to capture both short and long dependencies. The ConvLSTM model was chosen for its noise resistance, while the BiLSTM model was selected for its bidirectional sequential information review capability.

The comparison of training loss curves across LSTM, GRU, ConvLSTM, and BiLSTM models showcases the differences in their performance over epochs (Fig. 3b). These curves demonstrate how the training loss decreases with each epoch, highlighting the models' ability to converge and efficiently learn to estimate drop width. A minimal discrepancy between training and validation losses typically indicates that the model has captured the dataset's essential patterns without memorizing the data (overfitting) or failing to learn sufficiently (underfitting). This equilibrium suggests the model's effective generalization to unseen data, reflecting a successful training process. The validation loss is slightly lower than the training loss likely due to the use of dropout during the training process.

We compared regression algorithms and MSA methods for drop sliding based on our test set. The accuracy of MSA methods was significantly better than that of regression in estimating drop width (Table 1). The best result for the regression algorithms in terms of both RMSE and MAE criteria was obtained using random forest, with values of 109.0 µm and 90.0 µm, respectively. LSTM as an MSA algorithm achieved the best result based on both criteria with values of 67.6 µm and 58.5 µm, respectively. The 67.6 µm error would be translated to about 2.4% error percentage when considering the full range of drop width values in the test dataset. The evaluation of MSA methods took place by examining the accuracies across different sliding window sizes, 5, 10, 15, and 20. Using a sliding window size of 5, for instance, we will estimate one drop width using five frames. The results indicate that increasing the sliding window size leads to accuracy improvement. These findings provide evidence that temporal dependencies play a vital role in predicting drop width.

The main differences between random forest (RF) and LSTM with a sliding window size of 20 frames have been explored by visualizing their respective estimations on a small subset of the test dataset (Fig. 4a). For the sliding of a drop on a sample without defect and the D-Bl-1000 sample, the estimation accuracy of RF was found to be close to that of LSTM (Fig. 4a I, II). However, with an increase in the length of the block defect, the accuracy of RF gradually diminished (Fig. 4a III, IV). When estimating the drop width on samples with block defects D-Bl-2000 also D-Bl-3000, the RF achieved RMSE of 114.5 µm and 227.6 µm, while LSTM attained RMSE of 50.4 µm and 82.8 µm, respectively.

The diagrams that were discussed represented a subset of the test data. To assess the accuracy of the algorithms across the entire test dataset, we considered the measured/estimated drop width. We plot the measured width against the predicted value for every frame in the test dataset (Fig. 4b). The plot indicates that the LSTM model yields superior results compared to the RF model for frames where the drop width becomes < 3.5 µm. These

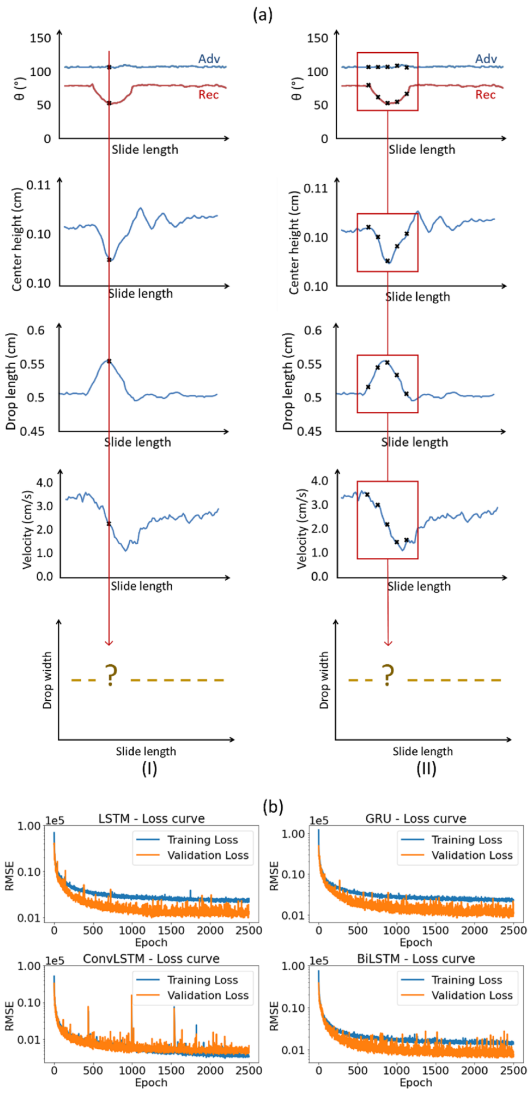


Figure 3. (**a**) A representation of two approaches to estimating the drop width. A regression approach that treats each observation independently (i). (**b**) An MSA approach that considers temporal dependencies by utilizing a sliding window (ii). The plotted θ_a and θ_r, drop center height, drop length, and velocity are based on a real measurement of a sample with a cylindrical defect. The calculation accounts for the middle line angle, which has not been explicitly depicted here for simplicity. (**b**) Comparison of training loss curves for LSTM, GRU, ConvLSTM, and BiLSTM architectures. The curves depict the evolution of the training loss over epochs, illustrating the convergence behavior and efficiency of each model in learning the task of estimating drop width. Note that the y-axis employs a logarithmic scale, necessitating careful examination of the data.

widths are present in cases where drop interacts with the surface defects. The dense clustering of scatter plot points around the line of identity in the LSTM graph indicates enhanced prediction accuracy for drop widths.

To better understand why RF is not good at predicting drop width in the presence of defects, we compared results obtained with LSTM's RMSE and that of RF (Fig. 4c). We separated videos of sliding drops on surfaces containing defects (red dots) and videos of sliding drops on surfaces without defects (black dots). It turned out that on samples without defects, the accuracy of both methods is similar. On samples with defects, LSTM is better.

The primary cause for a larger error in the RF model is its failure to precisely predict the drop's width while it is interacting with defects. This failure arises from regression models' failure to account for occurrence dependencies, resulting in the omission of crucial information. However, the RF results outperformed other regression

Type of model	Model	Sliding window size (# frames)	RMSE (μm)	MAE (μm)
Regression	Multilayer perceptron	–	144.8	122.4
	Gradient boosting	–	125.0	94.4
	Random forest	–	**109.0**	**90.0**
	Linear regression	–	194.9	179.3
MSA	GRU	5	93.2	83.9
	GRU	10	83.3	75.4
	GRU	15	83.3	74.9
	GRU	20	81.7	71.9
	LSTM	5	96.5	84.8
	LSTM	10	81.4	73.8
	LSTM	15	67.7	60.5
	LSTM	20	**67.6**	**58.5**
	ConvLSTM	20	77.8	64.3
	BiLSTM	20	71.1	62.8

Table 1. Exploring the accuracy of drop width estimation through regression and MSA. Significant values are in [bold].

algorithms due to the dataset's imbalance. This implies that the occurrence of defects is an infrequent event. The RF, as a bagging method, exhibits a reasonable level of robustness to imbalanced data by itself[57,58]. Nevertheless, for research studies of the dynamic behavior of drops on samples without defects[3] RF provides accurate width values. Even a simple linear regression with the below formula works on the defect-free part of our test dataset with 95.8 RMSE for 32 μl drops.

$$Dropwidth = -351.7 \times \theta_a + 411.6 \times \theta_r + 118.4 \times \text{D.L.} - 188.0 \times \text{D.C.H.} - 841.4 \times \text{Vel.} + 850.6 \times \text{M.L.A.} + 3351.9 \quad (6)$$

while θ_a is advancing angle, θ_r is receding angle, D.L. is drop length, D.C.H. is drop center height, Vel. is velocity, and M.L.A. is middle line angle.

Validation of a new sample

To validate our model, we produced a sample of PFOTS_Si containing a block defect, thickness = 800 μm, length = 3000 μm, and height = 23 μm. There were no videos related to this specific defect in neither the training nor testing dataset. The large size of this defect for the LSTM model poses the greatest difficulty in terms of prediction among all defects we considered. The defect of the new sample has a height of 23 μm, a dimension not included in the model's training data (as per SI-Data distribution). We intend to test this specific sample with defect to evaluate the model's ability to generalize. Following the same procedures as before, mirrors were employed to directly measure the drop width, allowing for a precise assessment of the prediction accuracy. As a result of mirror installation, the side-view visible slide length decreased to ≈3.5 cm.

We employed the obtained measurements to estimate the drop width through the developed LSTM model with sliding window size of 20 frames. It was only possible to directly measure the drop width, as done before, for the last ≈1.5 cm of sliding motion. This experiment shows that the developed model is not restricted to the final ≈1.5 cm of sliding motion, and the estimation for block defects is displayed for ≈3.5 cm of the slide. The drop width was measured and estimated for sliding drops at tilt angles of 42° (Fig. 5a) and 45° (Fig. 5b).

The tilt angle of 42° was the lowest angle at which the drop could slide and pass the defect. Thus, drop velocity was slow and the receding contact line pinned to the defect for a while (Fig. 5a, the red region of diagrams). By increasing the tilt angle, the drop velocity increased and the drop did not pin. The advancing contact angle was more affected than the lower tilt angle (Fig. 5b, the blue region of diagrams). Also, drop center height, drop length, and middle angle degree measures were used to estimate the drop width. The drop width estimation accuracy for videos of sliding drop at 42° was 112.5 μm, and at 45° was 86.5 μm.

Sensitivity analysis

To ascertain which variable(s) exert the greatest impact on the estimation of drop width, we carried out a feature importance analysis for the LSTM model with sliding window size of 20 frames. Thus, we utilized the gradient-based feature importance. The feature importance analysis reveals that drop width estimation is strongly influenced by the drop length, followed by the height of the drop's center and velocity (Fig. 6a). The velocity of the drop is an important factor as it determines the kinetic energy of the drop which affects the deformation and spreading of the drop. The drop length and the height of the drop play a crucial role in determining the drop width. The lower importance of other variables may be attributed to their high correlation with these primary features (SI-Correlation matrix).

Our main goal is to ascertain the significance of features to implement the developed model on external data. Hence, we visualized the data abundance of our dataset (Fig. 6b**, Violin graphs**). In the data abundance diagram, each point corresponds to an existing frame in our dataset, associated with a particular value on the y-axis. For example, in the drop length diagram, the concentration of data was observed from 0.42 cm to 0.60 cm. However,

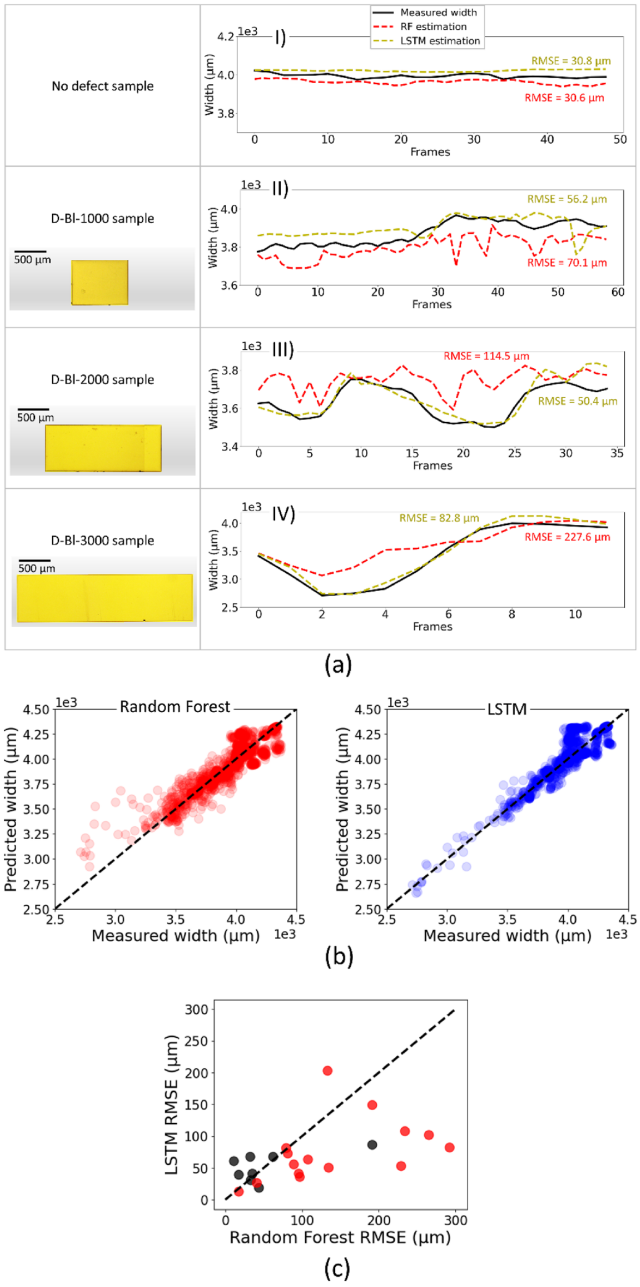


Figure 4. (**a**) Comparison of estimated and measured drop width. The representative examples were analyzed using both RF and LSTM (with 20 sliding window size) models on samples with and without defects. One sample without defect, and three samples with block defects, D-Bl-1000 (thickness = 800 µm, length = 1000 µm, and height = 106 µm), D-Bl-2000 (thickness = 800 µm, length = 2000 µm, and height = 74 µm), D-Bl-3000 (thickness = 800 µm, length = 3000 µm, and height = 174 µm), respectively. The drop width on the last ≈1.5 cm of the sliding movement is analyzed. In all cases defect microscopic image has been illustrated in the left column. (**b**) LSTM and RF predictions vs. real measurements on the entire test dataset based on frames. (**c**) Accuracy of LSTM vs. RF predictions on entire test dataset based on videos. The red dots represent the RMSE related to videos of sliding drops on surfaces containing defects. Also, black dots represent the same for surfaces without defects.

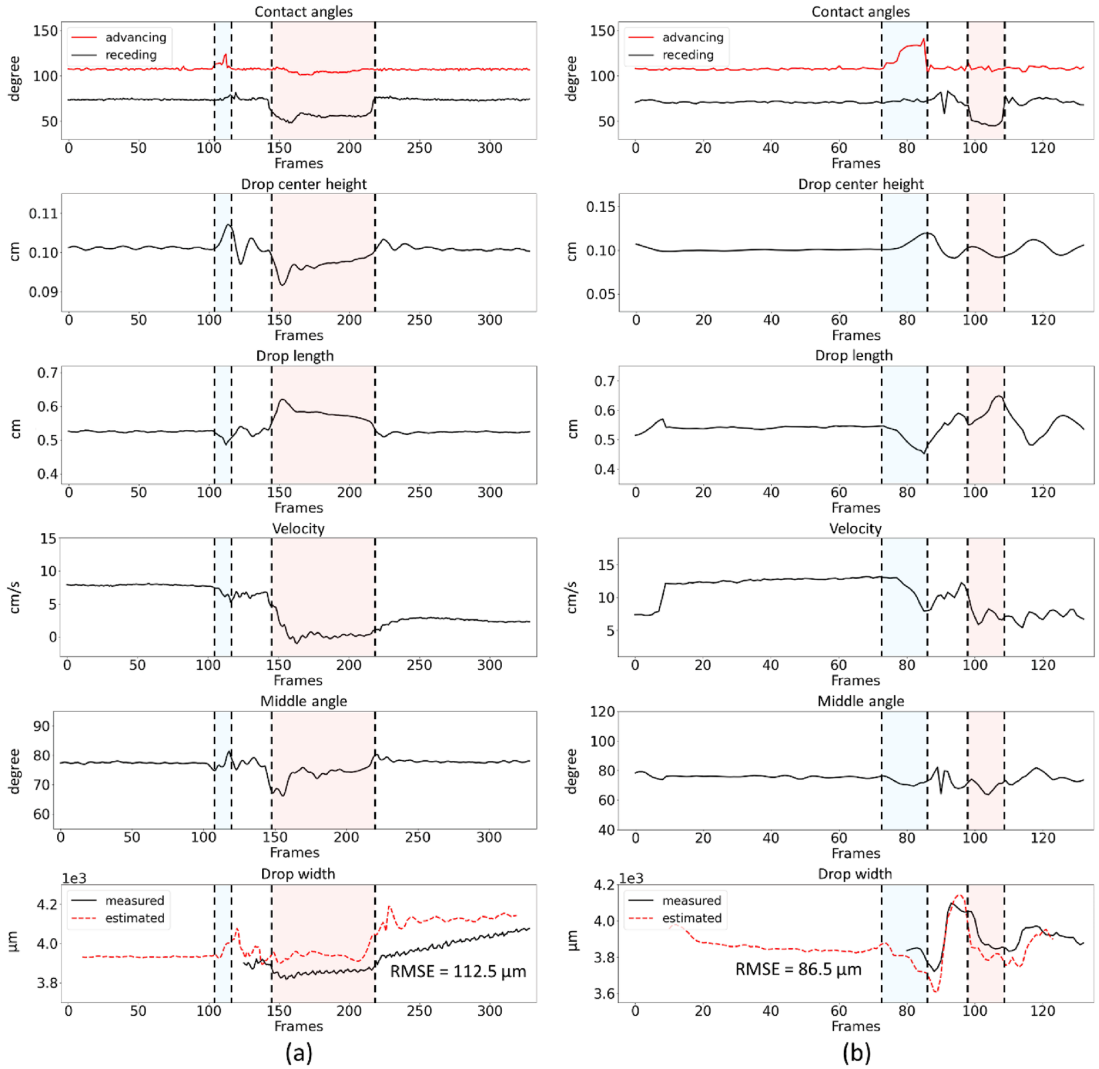


Figure 5. All side-views extracted measures and the estimated vs. measured drop width diagrams for drop sliding at (**a**) 42° and (**b**) 45° tilt angles. To showcase that the developed model is not restricted to the final ≈1.5 cm of sliding motion, the estimation of drop width was displayed for ≈3.5 cm of the slide (red dotted diagrams). The blue region represents when the advancing part was stuck to the defect. The red region represents when the receding part was stuck to the defect. The recording speed was 500 fps. The drop volume was 32 μl. In drop width estimating using LSTM with 20 sliding window size, we would lose the first 10 frames and the last 9 frames due to the sliding window size.

there were also data available within the range of 0.35–0.75 cm as well. Considering the significant role of drop length in width estimation, we suggest using drops that are between 0.35 and 0.75 cm length range as the model input to ensure high precision. In the same line, the drop center height shall be from 0.07 to 0.115 cm and the velocity span from ~ 0 up to 35 cm/s. The feature importance and data abundance analysis indicates that the developed LSTM method estimates the drop width with a negligible error. However, this negligible error value can only be realized when the main features of new measurements fall within the trained data range specified above. The model trained is expected to keep its accuracy on new surfaces that have physical defects, as long as the range of side-view measurements falls within the range we have reported. Please note, the model's ability to accurately estimate drop width out of the given range has not been investigated.

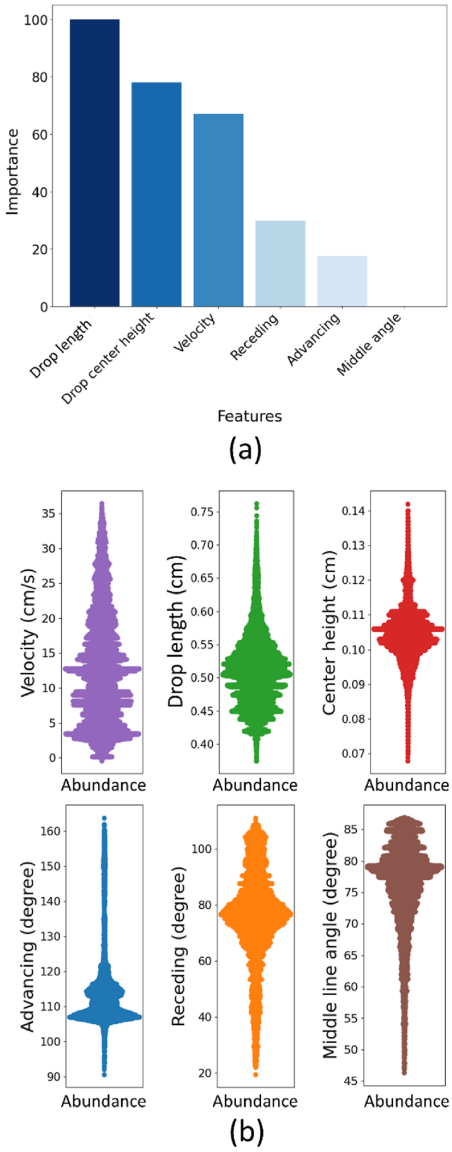


Figure 6. (**a**) Analysis of LSTM feature importance based on the Grad-CAM Method. The results show that drop length, drop center height, and velocity are the most important features in estimating drop width. The importance is normalized between 0 and 100. (**b**) The distribution of observations in the training data based on features. Each point represents an existing frame in the dataset, related to a particular value on the y-axis.

Conclusions

Including the temporal dependencies is critical for accurate estimations of drop width, when there is a defect on the surface that impedes drop motion. Our evaluation revealed that the LSTM model surpassed the competing models in terms of accuracy. With a result of 67.6 µm for the RMSE, the LSTM model outperformed the RF model, which achieved a best result of 109.0 µm. Taking into account the full range of drop width measurements in our dataset, which spans from a minimum of 1.6 mm to a maximum of 4.4 mm, the RMSE of 67 µm achieved by the LSTM model translates to an error percentage of 2.4%. The performance of the LSTM model eliminates the requirement for extra equipment while facilitating precise estimation along the entire sliding path. Thus the estimation of the drop width allows for a comprehensive analysis of drop behaviour.

Furthermore, the higher error observed in the RF model was attributed to the presence of relatively large defects on the surface. On smooth and homogeneous samples without defects, one can use a simple and interpretable regression algorithm like RF to determine drop width.

We've taken steps to enhance the model's versatility by exploring variations in drop viscosity and surface chemistry. As a result, the model's primary strength is its suitability for researchers working under similar conditions. We emphasize the importance of our sensitivity analysis section, which indicates boundaries for our model's applicability in various research endeavours.

In essence, the proposed models offer a novel approach for estimating front-view drop width solely based on side-view measures. This approach simplifies sliding drop analysis significantly and enables researchers to measure drop width along the entire sliding path. The latter is experimentally unattainable with an optical setup. According to our findings, the RF model is suitable for flat samples, while the LSTM model is preferable for samples with defects. As we pioneer this research field, we employed the most known models to establish a foundational framework for future research.

As next steps, to improve precision, generality, and stability in future research, collecting more extensive datasets and using more sophisticated models like transformers could be beneficial. The superior modelling features of transformers might unlock a more profound insight into the time series' temporal patterns. Moreover, the direct analysis of drop videos through the use of Convolutional Neural Networks (CNNs) might be beneficial.

Data availability

The supporting materials and data generated and analysed during this study are included in this published article. The dataset: The "Dataset.xlsx" represents the dataset we compiled after processing and integrating the sliding drop videos. Training and validation process: The "Training and validation process.ipynb" file provides a detailed, step-by-step explanation of how we trained the LSTM model with a 20-slide window, which was determined to be the best model based on RMSE. LSTM learning process: The "LSTM learning process.xlsx" file includes a representation of the learning process for the LSTM model utilizing a 20-slide window. LSTM weights: The "LSTM weights.h5" file represents the fully trained 20-slide window LSTM model that can be employed by others for the purpose of estimating drop width in a same condition.

Received: 25 October 2023; Accepted: 14 May 2024
Published online: 27 May 2024

References

1. Sbragaglia, M. *et al.* Sliding drops across alternating hydrophobic and hydrophilic stripes. *Phys. Rev. E* **89**(1), 012406 (2014).
2. Yonemoto, Y., Suzuki, S., Uenomachi, S. & Kunugi, T. Sliding behaviour of water-ethanol mixture droplets on inclined low-surface-energy solid. *Int. J. Heat Mass Transf.* **120**, 1315–1324 (2018).
3. Li, X. *et al.* Kinetic drop friction. *Nat. Commun.* **14**(1), 4571 (2023).
4. Extrand, C. W. & Gent, A. N. Retention of liquid drops by solid surfaces. *J. Colloid Interface Sci.* **138**(2), 431–442 (1990).
5. Extrand, C. W. & Kumagai, Y. Liquid drops on an inclined plane: The relation between contact angles, drop shape, and retentive force. *J. Colloid Interface Sci.* **170**(2), 515–521 (1995).
6. Lhermerout, R. & Davitt, K. Controlled defects to link wetting properties to surface heterogeneity. *Soft Matter.* **14**(42), 8643–8650 (2018).
7. Dussan, V. E. B. & Dussan, V. E. B. On the ability of drops to stick to surfaces of solids. Part 3. The influences of the motion of the surrounding fluid on dislodging drops. *J. Fluid Mech.* **174**, 381–397 (1987).
8. Antonini, C., Carmona, F. J., Pierce, E., Marengo, M. & Amirfazli, A. General methodology for evaluating the adhesion force of drops and bubbles on solid surfaces. *Langmuir* **25**(11), 6143–6154 (2009).
9. Pilat, D. W. *et al.* Dynamic measurement of the force required to move a liquid drop on a solid surface. *Langmuir* **28**(49), 16812–16820 (2012).
10. Tadmor, R. *et al.* Measurement of lateral adhesion forces at the interface between a liquid drop and a substrate. *Phys. Rev. Lett.* **103**(26), 266101 (2009).
11. Furmidge, C. G. L. Studies at phase interfaces. I. The sliding of liquid drops on solid surfaces and a theory for spray retention. *J. Colloid Sci.* **17**(4), 309–324 (1962).
12. Larkin, B. K. Numerical solution of the equation of capillarity. *J. Colloid Interface Sci.* **23**(3), 305–312 (1967).
13. Frenkel, Y.I. On the behavior of liquid drops on a solid surface. 1. The sliding of drops on an inclined surface. arXiv:physics/0503051 (2005).
14. Buzágh, A. & Wolfram, E. Bestimmung der Haftfähigkeit von Flüssigkeiten an festen Körpern mit der Abreißwinkelmethode. II. *Kolloid-Zeitschrift.* **157**, 50–53 (1958).
15. Gao, N. *et al.* How drops start sliding over solid surfaces. *Nat. Phys.* **14**(2), 191–196 (2018).
16. Li, X. *et al.* Spontaneous charging affects the motion of sliding drops. *Nat. Phys.* **18**(6), 713–719 (2022).
17. Yang, S., Hou, Y., Shang, Y. & Zhong, X. BPNN and CNN-based AI modeling of spreading and icing pattern of a water droplet impact on a supercooled surface. *AIP Adv.* **12**(4), 045209 (2022).
18. Rabbani, A. *et al.* DeepAngle: Fast calculation of contact angles in tomography images using deep learning. *Geoenergy Sci. Eng.* **227**, 211807 (2023).
19. Ni, P., Goto, H., Nakamoto, M. & Tanaka, T. Neural network modelling on contact angles of liquid metals and oxide ceramics. *ISIJ Int.* **60**(8), 1586–1595 (2020).
20. Kabir, H. & Garg, N. Machine learning enabled orthogonal camera goniometry for accurate and robust contact angle measurements. *Sci. Rep.* **13**(1), 1497 (2023).
21. Shumaly, S. *et al.* Deep learning to analyze sliding drops. *Langmuir* **39**(3), 1111–1122 (2023).
22. Shumaly, S. Drop width estimation. 2024 Available from: github.com/AK-Berger/Drop_width_estimation.
23. Ai, Y. *et al.* A deep learning approach on short-term spatiotemporal distribution forecasting of dockless bike-sharing system. *Neural Comput. Appl.* **31**(5), 1665–1677 (2019).
24. Zou, W. & Xia, Y. Back propagation bidirectional extreme learning machine for traffic flow time series prediction. *Neural Comput. Appl.* **31**(11), 7401–7414 (2019).
25. Shen, Z., Bao, W. & Huang, D.-S. Recurrent neural network for predicting transcription factor binding sites. *Sci. Rep.* **8**(1), 15270 (2018).

26. Che, Z., Purushotham, S., Cho, K., Sontag, D. & Liu, Y. Recurrent neural networks for multivariate time series with missing values. *Sci. Rep.* **8**(1), 6085 (2018).
27. Chien, Y.-W. *et al.* An automatic assessment system for Alzheimer's disease based on speech using feature sequence generator and recurrent neural network. *Sci. Rep.* **9**(1), 19597 (2019).
28. Hochreiter, S., Bengio, Y., Frasconi, P. & Schmidhuber, J. *Gradient flow in recurrent nets: the difficulty of learning long-term dependencies* (IEEE Press, New York, 2001).
29. Basodi, S., Ji, C., Zhang, H. & Pan, Y. Gradient amplification: An efficient way to train deep neural networks. *Big Data Min. Anal.* **3**(3), 196–207 (2020).
30. Rao, G., Huang, W., Feng, Z. & Cong, Q. LSTM with sentence representations for document-level sentiment classification. *Neurocomputing* **308**, 49–57 (2018).
31. Hochreiter, S. & Schmidhuber, J. Long short-term memory. *Neural Comput.* **9**(8), 1735–1780 (1997).
32. Hochreiter, S. & Schmidhuber, J. LSTM can solve hard long time lag problems. *Adv. Neural Inf. Process. Syst.* **9**, 1735–1780 (1996).
33. Greff, K., Srivastava, R. K., Koutník, J., Steunebrink, B. R. & Schmidhuber, J. LSTM: A search space odyssey. *IEEE Trans. Neural Netw. Learn. Syst.* **28**(10), 2222–2232 (2016).
34. Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. Learning phrase representations using RNN encoder-decoder for statistical machine translation. arXiv preprint arXiv:14061078. (2014).
35. Chung, J., Gulcehre, C., Cho, K., & Bengio, Y. Empirical evaluation of gated recurrent neural networks on sequence modeling. arXiv preprint arXiv:14123555. (2014).
36. Xu, G., Meng, Y., Qiu, X., Yu, Z. & Wu, X. Sentiment analysis of comment texts based on BiLSTM. *IEEE Access.* **7**, 51522–51532 (2019).
37. Siami-Namini, S., Tavakoli, N., Namin, A.S. The performance of LSTM and BiLSTM in forecasting time series. *Proc. IEEE.* (2019).
38. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S.,& Anguelov, D., et al. Going deeper with convolutions. (2015).
39. Gamboa, J.C.B. Deep learning for time-series analysis. arXiv preprint arXiv:170101887. (2017).
40. Bengio, Y., Courville, A. & Vincent, P. Representation learning: A review and new perspectives. *IEEE Trans. Pattern Anal. Mach. Intell.* **35**(8), 1798–1828 (2013).
41. Livieris, I. E., Pintelas, E. & Pintelas, P. A CNN–LSTM model for gold price time-series forecasting. *Neural Comput. Appl.* **32**(23), 17351–17360 (2020).
42. Xue, N., Triguero, I., Figueredo, G.P., Landa-Silva, D., Evolving deep CNN-LSTMs for inventory time series prediction. *Proc. IEEE.* (2019).
43. Kim, T.-Y. & Cho, S.-B. Predicting residential energy consumption using CNN-LSTM neural networks. *Energy* **182**, 72–81 (2019).
44. Ribeiro, M.T., Singh, S., Guestrin, C., Why should i trust you? Explaining the predictions of any classifier. *Proc. of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*. (2016).
45. Lundberg, S.M. & Lee, S.-I. A unified approach to interpreting model predictions. *Proc. Advances in neural information processing systems*. 30. (2017).
46. Zhou, B., Khosla, A., Lapedriza, A., Oliva, A.,& Torralba, A., Learning deep features for discriminative localization. *Proc. of the IEEE conference on computer vision and pattern recognition*. (2016).
47. Selvaraju, R.R., Das, A., Vedantam, R., Cogswell, M., Parikh, D., Batra, D. Grad-CAM: Why did you say that? arXiv preprint arXiv:161107450. (2016).
48. Gorski, L., Ramakrishna, S., Nowosielski, JM. Towards grad-cam based explainability in a legal text processing pipeline. arXiv preprint arXiv:201209603. (2020).
49. Kim, J., Oh, J. & Heo, T.-Y. Acoustic scene classification and visualization of beehive sounds using machine learning algorithms and Grad-CAM. *Math. Prob. Eng.* **2021**, 1–13 (2021).
50. Chao, Q., Wei, X., Tao, J., Liu, C. & Wang, Y. Cavitation recognition of axial piston pumps in noisy environment based on Grad-CAM visualization technique. *CAAI Trans. Intell. Technol.* **8**(1), 206–218 (2022).
51. Shumaly, S. 4S-SROF toolkit https://github.com/AK-Berger/4S-SROF2023
52. Liashchynskyi, P., & Liashchynskyi, P. Grid search, random search, genetic algorithm: A big comparison for NAS. arXiv preprint arXiv:191206059. (2019).
53. Freedman, D. A. *Statistical Models: Theory and Practice* (Cambridge University Press, 2009).
54. Haykin, S. *Neural Networks: A Comprehensive Foundation* (Prentice Hall PTR, 1998).
55. Ho, T.K. Random decision forests. *Proc. IEEE.* (1995).
56. Friedman, J. H. Greedy function approximation: A gradient boosting machine. *Ann. Stat.* **29**, 1189–1232 (2001).
57. Dittman, D.J., Khoshgoftaar, T.M., Napolitano, A. The effect of data sampling when using random forest on imbalanced bioinformatics data. *Proc. IEEE.* (2015).
58. Shumaly, S., Neysaryan, P., Guo, Y. Handling class imbalance in customer churn prediction in telecom sector using sampling techniques, bagging and boosting trees. *Proc. IEEE.* (2020).

Acknowledgements

We thank Andreas Best for measuring the size of the defects, also Gabriele Schaefer to make samples. We acknowledge financial support from Max Planck Center on Complex Fluid Dynamics (S.S.), and the Priority Programme 2171 Dynamic wetting of flexible, adaptive, and switchable surfaces (Grant No. BU 1556/36 and BE 3286/6-1: H.-J.B., R.B., X.L.). We acknowledge financial support by the German Research Society via the CRC 1194 (Project-ID 265191195) "Interaction between Transport and Wetting Processes", projects C07N (R.B., H.-J.B.). This project has received also funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (Grant Agreement No. 883631) (F.D., X.L., H.-J.B.).

Author contributions

S.S.: Conducting the methodology, data processing, machine learning coding, analyzing data, sliding drop experiments, preparing the dataset, writing the manuscript, and discussing the results regularly. F.D.: Conducting the methodology, data processing, analyzing data, producing samples, preparing the dataset, writing the manuscript, and discussing the results regularly. X.L.: Conducting the methodology, analyzing data, sliding drop experiments, preparing the dataset, and discussing the results regularly. O.K.: Conducting the methodology, analyzing data, writing the manuscript, and discussing the results regularly. W.S.: Conducting the methodology, analyzing data, and discussing the results regularly. Y.G.: Conducting the methodology, analyzing data, and discussing the results regularly. H-J.B.: Conducting the methodology, analyzing data, and discussing the results regularly. R.B.: Conducting the methodology, analyzing data, writing the manuscript, and discussing the results regularly.

Funding
Open Access funding enabled and organized by Projekt DEAL.

Competing interests
The authors declare no competing interests.

Additional information
Supplementary Information The online version contains supplementary material available at https://doi.org/10.1038/s41598-024-62194-w.

Correspondence and requests for materials should be addressed to R.B.

Reprints and permissions information is available at www.nature.com/reprints.

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

4.2 Supporting information

Estimating sliding drop width via side-view features using recurrent neural networks

*Sajjad Shumaly[1], Fahimeh Darvish[1], Xiaomei Li[1], Oleksandra Kukharenko[1], Werner Steffen[1], Yanhui Guo[2], Hans-Jürgen Butt[1], Rüdiger Berger[1]**

[1]Max Planck Institute for Polymer Research, Ackermannweg 10, D-55128, Mainz, Germany

[2]Department of Computer Science, University of Illinois Springfield, Springfield, IL, USA

* Corresponding Author. Email: berger@mpip-mainz.mpg.de

KEYWORDS: Sliding drop, Drop width estimation, Multivariate sequence analysis, Convolutional neural network (CNN), Long short-term memory (LSTM), Gated recurrent unit (GRU), bidirectional LSTM (BiLSTM), ConvLSTM

RNNs' architectures.

RNNs maintain a hidden state that acts as a memory (h), allowing them to capture and remember information from previous elements in the sequence (**Figure 1a**). It takes input at each time step, updates the hidden state, and produces an output. The RNN cell's formula is as follows:

$$h_t = tanh\,(W^{hx}x_t + W^{hh}h_{t-1} + b^h) \quad (1)$$

Where t denotes the time step, x_t denotes the current input, W is the weight, b is the bias, h_{t-1} and h_t denote the output of the last RNN cell, and current output, respectively (**Figure 1b**). The network's parameters, including weights and activation functions, are shared across all steps, enabling it to model sequential dependencies and relationships in the data. Learnable parameters, including weights and biases, determine how information is combined.

The LSTM cell's formula is as follows:

$$f_t = \sigma(W^f[h_{t-1}, x_t] + b^f) \quad (2)$$

$$i_t = \sigma(W^i[h_{t-1}, x_t] + b^i) \quad (3)$$

$$\tilde{c}_t = tanh(W^c[h_{t-1}, x_t] + b^c) \quad (4)$$

$$c_t = f_t \,.\, c_{t-1} + \, i_t \,.\, \tilde{c}_t \quad (5)$$

$$o_t = \sigma(W^o[h_{t-1}, x_t] + b^o) \quad (6)$$

$$h_t = o_t \,.\, tanh(c_t) \quad (7)$$

where c_{t-1}, and c_t denote last cell state, and current cell state, $\tilde{c}_t$ denotes candidate cell state, f_t denotes forget gate value, i_t denotes update gate value, o_t denotes output gate value, the operator '·' denotes the pointwise multiplication of two vectors (**Figure 1c**).

The GRU cell's formula is as follows:

$$z_t = \sigma(W^z[h_{t-1}, x_t] + b^z) \quad (8)$$

$$r_t = \sigma(W^r[h_{t-1}, x_t] + b^r) \quad (9)$$

$$\tilde{h}_t = tanh(W^h[r_t \,.\, h_{t-1}, x_t] + b^h) \quad (10)$$

$$h_t = \, z_t \,.\, \tilde{h}_t + (1 - z_t) \,.\, h_{t-1} \quad (11)$$

2

Where z_t denotes the update gate value, and r_t denotes the reset gate value (**Figure 1d**).

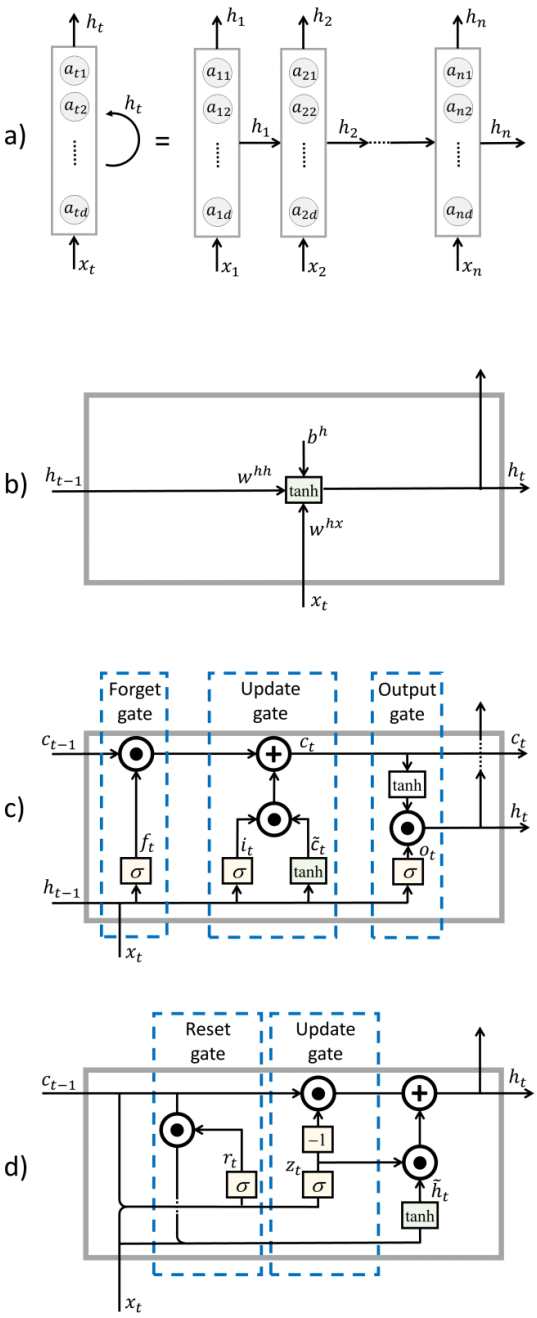


Figure 1. Visualizing the structure of RNN, LSTM, and GRU. a) Recurrent architecture representation. Left is shorthand notation and right is unfolded notation for RNNs. The d is the number of nodes that is

3

a hyper-parameter. b) A vanilla RNN cell's architecture. The figure provides a closer examination of an individual cell within the RNN architecture. c) An LSTM cell's architecture. d) A GRU cell's architecture.

4

Data distribution. We've gathered a dataset that includes both samples with defects and defect-free samples (**Figure 2**). To enhance the model's generality, the defects have varying geometries in each type. The defects themselves were generated using SU8 on silicon samples, as explained in the manuscript. Finally, all samples containing defects have been coated with PFOTS. For the defect-free samples, we've used different surfaces and applied various coatings.

Samples with defects:

- Block defect-I geometry: thickness = 800 μm, width = 1000 μm, height = 106 μm.
- Block defect-II geometry: thickness = 800 μm, width = 2000 μm, height = 74 μm.
- Block defect-III geometry: thickness = 800 μm, width = 3000 μm, height = 174 μm.
- Block defect-III geometry: thickness = 800 μm, width = 3000 μm, height = 23.0 μm. (The final validation sample)
- Cylindrical defect-I geometry: diameter = 800 μm, height = 31 μm.
- Cylindrical defect-II geometry: diameter = 800 μm, height = 47 μm.
- Step defect geometry: thickness = 800 μm, width = 2 cm, height = 30 μm.

Samples without defects:

- Thiols_Au: A perfluorodecanethiol monolayer on gold coated glass.
- PFOTS-Si: A 1H,1H,2H,2H-perfluoroctyltrichlorosilane coated silicon wafer sample.
- Glycerol on Thiols_Au: Varying glycerol concentrations ranging from 20% to 40% slid on Thiols_Au samples.
- Others: Twelve videos were recorded featuring various samples, including PFOTS coating on SiO2 and gold coating on glass.

5

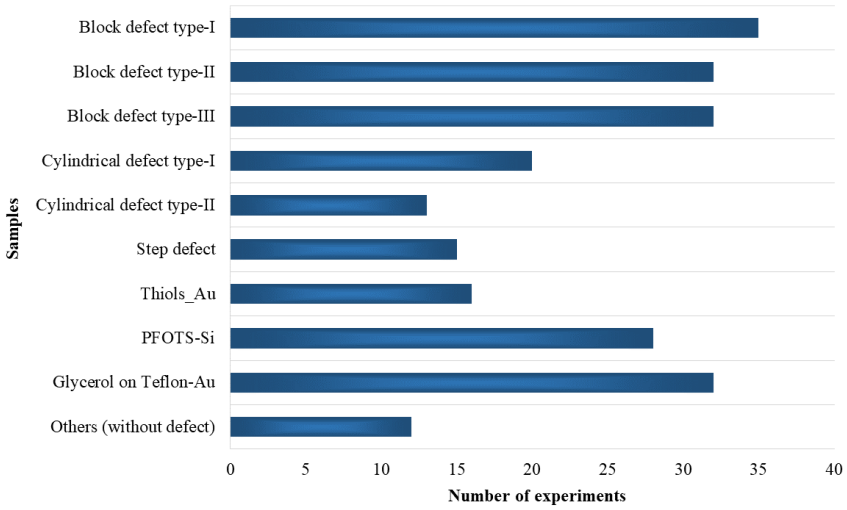


Figure 2. Data distribution.

6

Correlation matrix. The correlation matrix computed with Pearson correlation coefficients serves as a tool for quantifying and illustrating the linear relationships among the variables under investigation.

Table 1. The correlation matrix of variables.

	θ_a	θ_r	Drop length	Drop center height	Velocity	Middle angle degree
θ_a	**1.00**	0.19	-0.44	0.48	0.12	-0.06
θ_r	0.19	**1.00**	-0.76	0.48	0.08	0.75
Drop length	-0.44	-0.76	**1.00**	-0.68	0.09	-0.67
Drop center height	0.48	0.48	-0.68	**1.00**	-0.14	0.48
Velocity	0.12	0.08	0.09	-0.14	**1.00**	-0.23
Middle angle degree	-0.06	0.75	-0.67	0.48	-0.23	**1.00**

7

Data Availability

The supporting materials and data generated and analysed during this study are included in this published article.

- The dataset

  The “Dataset.xlsx” represents the dataset we compiled after processing and integrating the sliding drop videos. In this dataset, the "Status" column indicates whether a video is associated with training, testing, or final validation measurements. Initially, we made random selections for these assignments but later maintained consistency across all algorithms to ensure a fair comparison. It's worth noting that the final validation records differ from the regular validation records. After dividing the dataset into testing and training subsets, we further split the training data into the typical training and validation sets for the training process. The final validation involves measurements conducted externally to the dataset, serving to assess the model's validity.

- Training and validation process

  The "Training and validation process.ipynb" file provides a detailed, step-by-step explanation of how we trained the LSTM model with a 20-slide window, which was determined to be the best model based on RMSE. Using this file, reviewers can access the code, variables, and hyperparameters for examination. Furthermore, the document demonstrates how we utilized the trained model to incorporate the final validation metrics and estimate drop width. Ultimately, this file will be uploaded to GitHub and made freely accessible to everyone.

- LSTM learning process

  The "LSTM learning process.xlsx" file includes a representation of the learning process for the LSTM model utilizing a 20-slide window. The learning process is based on its loss (MSE). Also MAE metric during the learning process is accessible.

- LSTM weights

  The "LSTM weights.h5" file represents the fully trained 20-slide window LSTM model that can be employed by others for the purpose of estimating drop width in a same condition.

Chapter 5

Publication 3

5.1 CNN-Transformer with Absolute Positional Encoding Optimized for Low-Dimensional Inputs: Applied to Estimate Sliding Drop Width

5.1.1 Summary and author contribution

This publication builds on the insights and limitations identified in P2. Whereas P2 relies on hand-crafted side-view features produced by a separate processing tool, P3 moves to an end-to-end approach that operates directly on raw video data. A position-invariant spatiotemporal windowing strategy mitigates the fact that sliding drops occupy only a small region of each frame and avoids positional bias that can arise from naive cropping. The proposed architecture combines a compact VGG8-style backbone with ConvTran to efficiently capture both fine contour cues and long-range temporal dependencies. In addition, a low-dimensional absolute positional encoding is introduced to address a key bottleneck when applying transformer models to small feature spaces, yielding both theoretical motivation and empirical gains. Overall, P3 improves accuracy compared with P2 and enhances interpretability via attention-based analysis and Grad-CAM visualizations.

The author made a central contribution to the conception and design of the study, as well as the development of the deep-learning architecture. The author was responsible for implementing the model, performing the training and evaluation, and preparing the data and visualization of results. The author also contributed to the interpretation of findings, preparation of the open-source materials, and writing of the manuscript sections on methods and results.

5.1.2 Scientific publication

CNN-Transformer with Absolute Positional Encoding Optimized for Low-Dimensional Inputs: Applied to Estimate Sliding Drop Width

Sajjad Shumaly[1(✉)], Fahimeh Darvish[1], Mahsa Salehi[2], Navid Mohammadi Foumani[2], Oleksandra Kukharenko[1], Hans-Jürgen Butt[1], Ulrich Schwanecke[3], and Rüdiger Berger[1]

[1] Max Planck Institute for Polymer Research (MPI-P), Mainz, Germany
shumalys@mpip-mainz.mpg.de
[2] Department of Data Science and Artificial Intelligence, Monash University, Melbourne, VIC, Australia
[3] Department of Computer Science and Media, RheinMain University of Applied Sciences, Wiesbaden, Germany

Abstract. High-speed video recordings are crucial for investigating drop dynamics and their interactions with surfaces. Measuring the width of sliding drops, a key parameter linked to frictional forces, requires additional equipment like cameras or mirrors, complicating experimental setups and limiting observable areas. This study introduces a novel method that simplifies the measurement process by employing artificial neural networks to estimate millimeter-scale drop width directly from side-view video data. Our approach processes raw video footage to dynamically identify features most indicative of drop width. By treating drop behavior as an extrinsic time-series problem, our model effectively captures temporal dependencies in video sequences. We propose a VGG8-inspired architecture optimized for small and low information density video datasets. This architecture is combined with our novel position invariant video processing methodology that efficiently removes non-essential regions, reducing computation time by 84%. We further integrate ConvTran, a state-of-the-art time-series classification model, with an enhanced Absolute Position Encoding, improving the encoding's dot-product and lowering drop width estimation errors. Our novel neural network architecture achieved a root mean square error of 48 µm (1.7 % relative error), where each pixel corresponds to approximately 44 µm. Code and data are open-sourced at: https://github.com/shumaly/position_invariant_cnn_transformer.

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/978-3-032-06118-8_1.

I. Dutra et al. (Eds.): ECML PKDD 2025, LNAI 16021, pp. 3–21, 2026.
https://doi.org/10.1007/978-3-032-06118-8_1

Keywords: position invariant video processing · low-dimensional absolute positional encoding · extrinsic time series · spatiotemporal CNNTransformer

1 Introduction

Video analysis of sliding drops enables quantitative studies of sliding forces and liquid-solid interfacial properties [1,2]. Sliding forces depend on drop width [3,4]. A recent investigation by Li et al. focused on drops sliding down an inclined surface, presenting an empirical equation of the friction force F_f versus drop velocity U [3]:

$$F_f = F_0 + \beta w U \eta \tag{1}$$

Here, β is a dimensionless friction coefficient, w is the width of the drop while sliding, η is the viscosity of the liquid, and F_0 is the friction force extrapolated to velocity $U = 0$. The friction force of drops that just start sliding is described by the Furmidge equation [5–8]:

$$F = k\gamma w(\cos\theta_r - \cos\theta_a) \tag{2}$$

where γ is the liquid-air surface tension, θ_a is the advancing contact angle, θ_r is the receding contact angle, and k is a geometry factor [4,9]. The Furmidge equation also appears of frictional forces at low velocities [10]. The dynamic contact angles vary with velocity and can be easily measured from a side view.

Friction force is essential for detecting surface inhomogeneities, assessing interfacial stability, monitoring viscoelastic energy dissipation [43], and also is critical in anti-icing [41] and surface coating quality [42]. However, determining the drop width during a standard sliding drop experiment remains a challenging task. Adding cameras for bottom- or top-view measurements is not feasible since these views show the drop's central width, not the drop's contact line width. The drop's contact line width is narrower on surfaces with contact angles $> 90°$. Front-view imaging of drops is feasible by installing two mirrors or a second, time-synchronized high-speed camera [10,11]. However, it is limited to a sliding length of only ≈ 1.5 cm, as the drop moves toward the mirror and cannot stay within the camera's focus range for an extended period. To address these limitations, Shumaly et al. recently proposed a deep learning-based multivariate time-series analysis approach that leverages side-view measurements to estimate the front-view drop width, eliminating the need to add additional cameras or devices and without limiting sliding length [12].

Practical Significance. Previous research has relied on predefined measures extracted from side-view videos—such as contact angles, drop length, height, and the velocity of the drop's center—to estimate drop width. While these features are deemed important by existing literature, they may not capture all the nuanced interactions that occur, especially when drops encounter surface defects. When a drop moves over a surface with a single defect, its center velocity

decreases upon encountering a defect and increases after surpassing it (Fig. 1, black line). Meanwhile, the advancing and receding velocities exhibit distinct behaviors as they interact with surface defects in different ways (Fig. 1, red and blue lines). Monitoring only the center velocity fails to account for these differences, limiting estimation accuracy. The advancing and receding contact lines engage with the defect differently, revealing nuanced behaviors that are not captured when considering only the center velocity.

The gap in knowledge lies in the absence of a comprehensive method that can autonomously extract and prioritize relevant features from raw video data to describe the physics of sliding drops. Current models do not leverage the full potential of video data to identify subtle but important features that could enhance measurement precision, especially in challenging scenarios involving surface defects. Moreover, if we can automatically extract features, it will open up new opportunities to explore which segments of the drop contour line are crucial. For instance, we could investigate whether a combination of pixels in the drop's receding section or even the reflections within the drop itself might provide essential information on drop width (Fig. 1, green drop curves). Furthermore, this method enhances estimation accuracy and increase robustness against environmental variations, including optical distortions such as minor defocusing and focus irregularities, motion blur, lighting fluctuations, as well as dust within the lenses and scattered lights that cause noise in video frames.

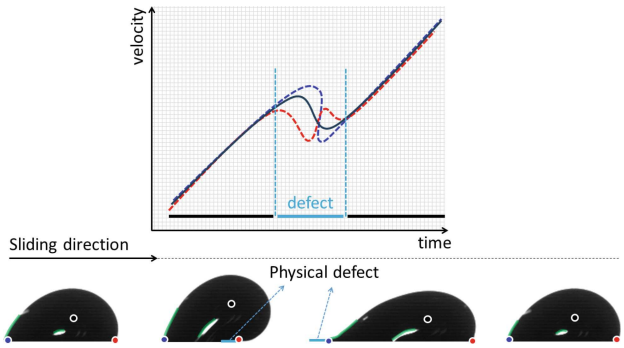


Fig. 1. Velocity profiles of a sliding drop over a surface defect. The diagram depicts the velocities at the drop's center (black), advancing edge (red), and receding edge (blue). Colors in the plot match the colored points on the schematic. As the drop interacts with the defect, the center velocity decreases, while the advancing and receding edges respond differently, revealing nuanced behaviors beyond center velocity analysis. Green curves indicate potential areas of interest for a more detailed investigation of drop dynamics. (Color figure online)

1.1 Main Contributions

In this study, we introduced a novel deep learning approach for accurately estimating the width of sliding drops directly from side-view video data. Key outcomes and advancements of our work include:

- **Position Invariant Video Processing:** Our proposed position invariant video processing method mitigates overfitting due to positional bias while significantly reducing computational load by approximately 84%. It is applicable to scientific problems involving the motion of small objects of interest, especially when data availability is limited.
- **Low-Dimensional Absolute Position Encoding:** Our proposed ldAPE effectively addresses the anisotropic limitations commonly encountered in conventional positional encoding methods for low-dimensional time-series data. Empirically, it outperforms both tAPE and Sin-APE on 32-dimensional data, with theoretical advantages extending up to 128 dimensions.
- **Optimized CNN-Transformer Architecture:** We developed a custom VGG8-inspired CNN architecture specifically designed for video datasets characterized by low information density. Coupled with the ConvTran time-series transformer, our model efficiently captures intricate spatiotemporal interactions. We achieved an RMSE of 48.4 μm, corresponding to a low error rate of just 1.7%. This demonstrates a considerable improvement over previous state-of-the-art models, especially in challenging scenarios involving surface defects.
- **Robustness and Interpretability:** Based on Grad-CAM visualizations, we confirmed that our model robustly identifies critical drop features, including subtle edges and reflections. This capability not only improves estimation accuracy but also enhances interpretability, offering insights into the underlying physics of drop-surface interactions.
- **Open Source Contribution:** To support future research and foster collaboration within the scientific community, we release our comprehensive sliding drop video dataset and the source code. This contribution enhances reproducibility, supports model inference, and promotes advancements in ML-based experimental fluid dynamics research.

2 Related Work

2.1 Machine Learning and Surface Science

The integration of machine learning into surface science enhances drop dynamics and contact angle analysis, improving complexity handling. Yancheshme et al. applied a random forest model to predict the behavior of impacting drops on hydrophobic and superhydrophobic surfaces [13]. Their goal was to determine the optimal conditions for inducing bouncing behavior during drop impact. They analyzed a broad set of predefined measures, including the drops' physical properties, kinematic characteristics, and surface attributes. Similarly, Zhang

et al. developed a method to optimize the contact angle on rice leaf surfaces by comparing artificial neural networks (ANN) and response surface methodology (RSM) [14]. They focused on factors such as temperature, humidity, and pesticide concentration to determine the best conditions for minimizing the contact angle. ANN outperformed RSM in contact angle prediction, with pesticide concentration as the key factor. Kokalis et al. proposed a method to classify composite insulators into hydrophobicity classes using convolutional neural networks (CNNs) [15]. They used a spray method to collect images and train CNNs for insulator classification, removing human subjectivity. In the same way, Roy et al. introduced a method for detecting the hydrophobicity grade of polymeric insulators using Bi-directional Long Short-Term Memory (Bi-LSTM) classifier [16]. Rabbani et al. employed two deep learning models with fully connected dense layers to predict contact angles in tomography images of porous materials [17]. Kabir et al. used ResNet-50 to estimate contact angles, overcoming fitting limitations on hydrophilic surfaces [18]. A recent deep learning study in surface science developed a method (4S-SROF), enabling systematic analysis of sliding drops, even when occupying a small image region [19]. Shumaly et al. introduced a method based on regressions and Recurrent Neural Networks (RNNs) to estimate sliding drop width using predefined side-view features [12]. Their Long Short-Term Memory (LSTM) model demonstrated the best performance, estimating sliding drop width with a low error of 2.4% (67.6 μ m RMSE), eliminating the need for cumbersome equipment while maintaining an unrestricted view of sliding drops. We now introduce more advanced end-to-end deep learning models capable of extracting features without relying on predefined physics-based measurements, enhancing accuracy to estimate sliding drop width.

2.2 Time Series Extrinsic Regression

Time series extrinsic regression (TSER) is a regression task aimed at understanding the relationship between a time series and continuous scalar variables. Although numerous papers are published annually on time series classification [20,21] and time series forecasting [22–24], time series extrinsic regression has received limited attention [25]. In this study, we address a TSER problem, reconstructing a time series (front-view) from a set of time series (side-view). Our approach employs a machine learning framework, formulating the task as a regression problem where the input consists of consecutive drop images and the output is a scalar value. Regression involves predicting a continuous numeric value based on a set of input features [26]. However, our goal is to estimate values that may extend the input time series or be indirect to it, without being restricted to future values.

Similar studies on regression involve estimating heart rate based on data gathered from accelerometers [27,28]. Random Convolutional Kernel Transform (ROCKET) has demonstrated state-of-the-art results in various time series tasks by leveraging a set of random convolutional kernels to extract informative features efficiently [29]. InceptionTime, a deep learning-based approach inspired by the Inception architecture, enhances feature extraction, making it effective

for capturing both short- and long-term temporal dependencies [30]. Similarly, Transformer for Time Series (TST) has been proposed as an attention-based model that excels in capturing intricate relationships within time series data by leveraging self-attention mechanisms [31]. ConvTran, a convolutional transformer model, has recently gained recognition. By combining convolutional feature extraction with transformer-based sequence modeling, ConvTran achieves superior performance in handling both local and global dependencies, making it particularly well-suited for tasks like TSER [32].

3 Materials and Methods

3.1 Data Collection

The sliding drop setup consists of a high-speed camera with a telecentric lens to record drop motion under uniform backlighting. Two parallel mirrors capture the front view by reflecting the backlight. The entire optical system is mounted on a rotatable breadboard to maintain alignment. Distilled water drops (32 μl) are deposited onto a tilted plane using a peristaltic pump connected to a grounded syringe needle. The technical details and a schematic of the setup and sample preparation are presented in Supplementary Information (SI) Sections S.1, and S.2. Installing the mirrors restricted the focus of the front-view camera to the last $\approx$1.5 cm of the slide path. Data was collected only within this region. Therefore, defects were fabricated on the last centimeter of the samples. The dataset was filtered to include videos with 20–250 frames for consistency. The dataset consists of 235 videos with a resolution of 1280×1024 pixels, containing a total of 11,944 frames. The number of frames per video varies depending on the drop velocity.

3.2 Data Augmentation

We applied data augmentation to mimic real-world imaging variations and enhance robustness. The techniques included brightness adjustment, Gaussian blur filtering, and artifact generation. Brightness adjustment varied image intensity by $\pm 15\%$ to account for ambient fluctuations. Gaussian blur was applied with randomly selected kernel sizes (1×1, 3×3, 5×5) to simulate defocusing and motion blur. Image artifacts were introduced as irregular stains and radiance spots to mimic lens smudges and reflections. Irregular stains were generated using sinusoidal perturbations on random circular shapes, followed by transformations such as stretching, rotation, and scaling. Radiance spots were simulated using Canny edge detection to localize drop edges, followed by circular gradient overlays. More details and pseudo-codes are provided in SI Section S.3.

3.3 Position Invariant Video Processing Methodology

Captured high-speed video frames of sliding drops have a resolution of 1280×1024 pixels. In our dataset, the largest drops reach 216×99 pixels. An initial approach involved cropping frames to 1280×99 pixels, preserving the drop's

horizontal path while removing unnecessary upper and lower portions (Fig. 2a). However, this approach introduced several challenges.

Firstly, the drop occupies only a small fraction of the cropped frame, leaving extensive empty space. Secondly, the model may overfit by associating drops with their absolute positions in the image rather than focusing on their shape and velocity, which are the relevant features. For instance, surface defects are always located in the last centimeter of the sliding path due to video capture constraints [12]. This carries the risk that the model becomes too closely adapted to the droplet's dynamic behavior at a specific location, thereby limiting its ability to generalize to defects appearing at other positions.

To address these challenges, we introduced a 3D sliding window centered on the drop, which we call the sliding spatiotemporal window (SSW). We set the window size to 216×99 pixels, matching the maximum observed drop dimensions (Fig. 2b). This window follows the drop's movement, keeping it centered in the frame and reducing irrelevant background. The impact of input tensor size on memory usage and computation time was obtained using a dummy input. It assesses the general computational footprint of the model's forward pass. The total memory usage M and total time T were computed as follows:

$$M = \sum_{i=1}^{n} S_i, \quad T = \sum_{i=1}^{n} t_i, \tag{3}$$

Here, S_i and t_i represent the memory usage (in bytes) and time (in milliseconds) of the i-th operation, respectively. Two experiments were conducted with different input tensor sizes: (216×99) notated as "SSW", and (1280×99) notated as "original". The percentage reduction in memory usage and computation time was computed as

$$\Delta M\% = \left(1 - \frac{M_{\text{ssw}}}{M_{\text{original}}}\right) \times 100\% = \left(1 - \frac{1796.8 \text{ MB}}{10153.8 \text{ MB}}\right) \times 100\% \approx 82.3\%, \tag{4}$$

$$\Delta T\% = \left(1 - \frac{T_{\text{ssw}}}{T_{\text{original}}}\right) \times 100\% = \left(1 - \frac{239.5 \text{ ms}}{1518.1 \text{ ms}}\right) \times 100\% \approx 84.2\%. \tag{5}$$

These results indicate that reducing the input tensor size led to an approximately 82% decrease in memory usage and an 84% decrease in computation time, while the number of model parameters remained unchanged.

Capturing temporal dynamics is essential for accurate drop width estimation. To track the drop's movement over time, we set the sequence length to 20 frames, meaning each model input consists of 20 consecutive frames with the drop centered within the SSW. Studies show that 20-frame sequences effectively capture key drop dynamics without overloading the model [12]. In general, frames 1 to 9 correspond to the past relative to the target frame (frame 10), whose width we aim to estimate, while frames 11 to 20 represent its future.

However, centering drop images inadvertently removes the drop's relative positional information within the sequence, which carries valuable temporal cues

about its motion. To retain motion cues, we tracked the drop's center relative to its start. However, directly including the drop's center position could lead the model to overfit to absolute drop locations. To avoid this, we incorporated the first derivative of the drop's position with respect to time, which corresponds to its velocity. We approximated the velocity using a first-order finite difference. Specifically, we calculated it as $v_t = (x_t - x_{t-1})/\Delta t$, where x_t is the horizontal position of the drop in frame t, and Δt is the time interval between frames. The resulting velocity time series was added as an input to the model. This helped us retain temporal motion cues while removing the risk of overfitting to absolute drop positions. Incorporating the velocity time series serves two key purposes. First, velocity is crucial for understanding drop dynamics, as it reflects frictional forces, surface interactions, and acceleration. Most importantly, with a fixed frame rate, velocity encodes positional changes and establishes a temporal link between frames.

Our approach ensures that the model focuses on the drop's shape and motion rather than its position. Additionally, it extracts only the drop region (216×99) from the original frame (1280×99), achieving an 84% reduction in computation time. We refer to this approach as position invariant video processing.

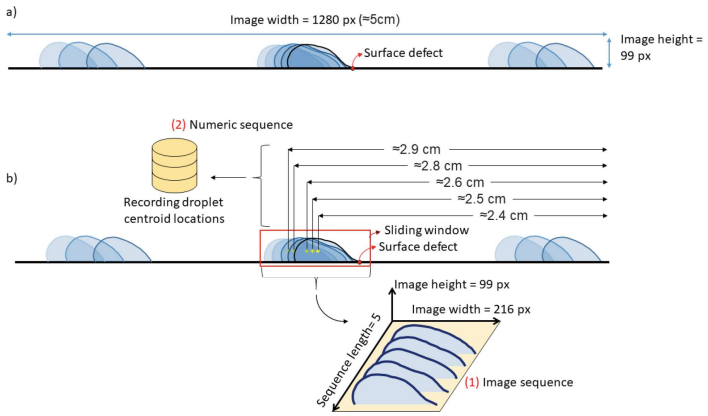


Fig. 2. Data preparation and pipeline for formatting input for the model. a) Initial approach: Cropping the full sliding path (1280×99) results in extensive empty space and positional bias due to the drop's varying location. b) Improved method: Using a SSW of size 216×99 pixels, matching the maximum drop dimensions. For demonstration, a 5-frame sequence is shown, while the model utilizes 20 frames for effective drop analysis.

3.4 Spatiotemporal Model Architecture

The model begins with a VGG-style 2D CNN to extract spatial features from consecutive video frames (Fig. 3). The architecture is adapted for smaller

datasets and images with lower informational density. It is inspired by VGG8, but replaces standard pooling layers with BlurPooling and employs the Gaussian Error Linear Unit (GELU) as the activation function. BlurPooling improves shift-equivariance, leading to better generalization [36]. It consists of four convolutional blocks with 64, 128, 256, and 512 filters, each featuring a 3×3 convolution (padding = 1). We refer to this architecture as BlurVGG8. The extracted spatial features are reshaped to align with the temporal data. Velocity from the 4S-SROF method [19] is processed through a fully connected layer for dimensional consistency before being integrated element-wise with spatial features. The position invariant video processing method stacks consecutive drop images, but to retain relative positional information, it requires integrating the velocity to preserve temporal dynamics.

The velocity encoded data is processed by the ConvTran architecture, starting temporal convolutional layers that refine short-term dependencies, followed by spatial convolutional layer. The embedding size is set to 64, followed by temporal convolutional layers that refine short-term dependencies. Next, position encoding is applied to enhance temporal awareness. We introduce an improved Absolute Position Encoding (APE), called low-dimensional Absolute Position Encodings (ldAPE), to enhance the model's capability. Simultaneously, efficient Relative Position Encoding (eRPE) captures relative frame distances. Since transformers process data in parallel, explicitly encoding temporal order is essential [38]. Next, the transformer encoder was applied with self-attention to capture long-range dependencies, analyzing frame interactions and tracking drop behavior. The number of heads was set to 4, and the feed-forward dimension was adjusted to 128 for our specific task.

We set the learning rate to 0.0001, used the AdamW optimizer with a weight decay of 1×10^{-5}, and selected a batch size of 16. We split the dataset into training, testing, and validation sets with a 60%/20%/20% distribution. The model was trained to minimize the Mean Squared Error (MSE) loss between predicted and actual widths. Performance was evaluated using the Root Mean Squared Error (RMSE) metric on the test set to maintain consistent units. To mitigate overfitting, the maximum training epochs were set to 400, with early stopping triggered by validation loss. All experiments were performed on a high-performance computing system with a single node featuring 36 CPU cores, 250 GB of memory, and an Nvidia A100 GPU.

3.5 Low-Dimensional Absolute Position Encodings

In transformer architectures, the self-attention mechanism alone cannot capture the natural order of sequential data. However, preserving the order of the sequence is crucial for accurate analysis, especially when dealing with time-series data. To overcome this limitation, transformer-based models introduce positional encoding, which injects order-related information into the input representation. The positional encoding ensures that the model can distinguish between different positions in the sequence and maintain the relationships. There are different

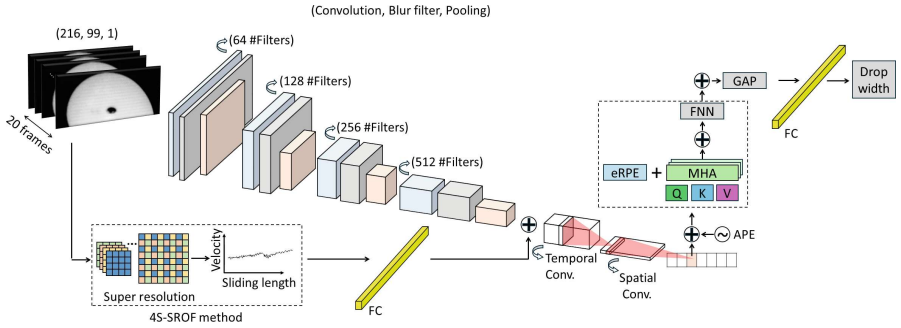


Fig. 3. Architecture of the spatiotemporal model. BlurVGG8 extracts spatial features using four convolutional blocks. Temporal dynamics are preserved by integrating velocity data with spatial features. The ConvTran architecture refines these features with additional convolutions, position encoding (APE and eRPE), and a Transformer encoder to capture long-range dependencies.

types of positional encoding such as absolute positional encoding (APE) and relative positional encoding (RPE) as the most common techniques [39, 40].

In the APE method, absolute position information is directly incorporated into the input embedding. This is achieved by adding a position-specific encoding to each input vector, formulated as:

$$x_i = x_i + p_i \tag{6}$$

Here, $p_i \in R^{d_{\text{model}}}$ represents the positional embedding corresponding to position i, and x_i denotes the input embedding at that position. d_{model} refers to the dimension of the model's hidden representations. The positional embedding is typically defined using sine and cosine functions as follows:

$$p_i(2k) = \sin(i\omega_k), \quad p_i(2k+1) = \cos(i\omega_k) \tag{7}$$

where

$$\omega_k = 10000^{-2k/d_{\text{model}}} \tag{8}$$

While i and k are both indices, i corresponds to the feature dimension, and k is the index of the frequency components. This method (called Sin-APE) has been widely used in transformer-based architectures [38]. Sin-APE was originally proposed for language modeling, where high embedding dimensions such as 512 or 1024 are typically used. However, it exhibits limitations when applied to time series data. In low embedding dimensions, the dot product between position encodings does not consistently decrease with increasing positional distance, leading to the loss of the distance awareness property. To address this issue, time Absolute Positional Encoding (tAPE) has been introduced [32]. This method modifies the frequency term to account for both the embedding dimension d_{model} and the sequence length L, ensuring a more balanced frequency distribution:

$$\omega_k^{\text{tAPE}} = \omega_k \cdot \frac{d_{\text{model}}}{L} \tag{9}$$

Here, L is the total length of the time series.

We modified the absolute positional encoding by adjusting the frequency term to improve accuracy. The new formulation is given by:

$$\omega_k^{\text{ldAPE}} = 35^{-2k/d_{\text{model}}} \cdot \frac{2\sqrt{d_{\text{model}}}}{L} \tag{10}$$

This adjustment introduces a scaling factor that accounts for both d_{model}, and L. By modifying ω_k, the encoding achieves a more balanced frequency distribution, and enhancing the model's ability to distinguish between positional embeddings. We refer to this method as low-dimensional Absolute Positional Encoding (ldAPE). The dot product between positional embeddings at a fixed reference position reflects their similarity. Compared to other methods, ldAPE produces a broader and more distinct, yet smooth and noise-free, distribution of similarity scores across positions (Fig. 4a), enhancing the model's ability to differentiate between them. Also, in ldAPE, the positional encodings for positions 2 and 20 show minimal overlap, indicating that ldAPE enhances position distinguishability more effectively than other methods (Fig. 4b-d). The ldAPE demonstrates a better dot product than the other mentioned APEs for dimensions below 128, SI Section S.4.

4 Results and Discussion

We tested LSTM models with 64, 128, and 256 units, as well as Bi-LSTM models with the same configurations. The Bi-LSTM models consistently outperformed the LSTM models, prompting us to use Bi-LSTM architectures for further experiments.

Initially, we tested transformer models and the VGG16 architecture, known for their effectiveness in capturing complex patterns and features across various tasks [34, 35]. However, due to the limited amount of training data available and low information density image frames, these models did not perform as well as expected (Table 1). The concept of low information density has been used to compare information density in computer vision and Natural Language Processing (NLP), suggesting that pixels in computer vision contain less information than words in NLP [37]. Additionally, different regions of an image contribute unequally to its overall meaning. Based on this, we argue that our images have even lower information density than typical computer vision images, as only the drop contour is relevant while the rest of the image holds minimal significance. To address this, we switched to VGG8, a streamlined version of the VGG architecture with lower complexity. This change achieved an RMSE of 63.54 μm, surpassing earlier studies that used features based on domain knowledge (RMSE of 67.6 μm [12]).

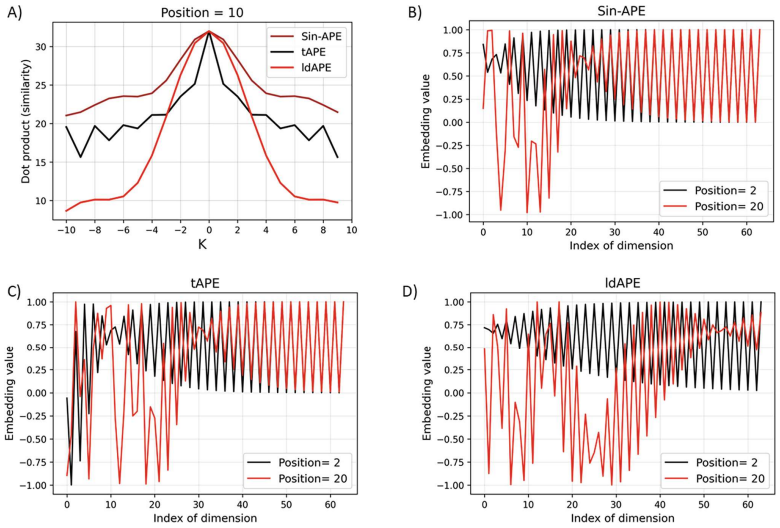


Fig. 4. Comparing different absolute positional embeddings. a) Dot product of absolute positional embeddings, demonstrating the wider similarity axis coverage in ldAPE with reduced fluctuations. K represents the relative distance between two positions b–d) Embedding values for positions 2 and 20 in a sequence of length 20 for Sin-APE, tAPE, and ldAPE, respectively, highlighting the improved position distinguishability in ldAPE.

Performance improved even more after modifying VGG8, replacing standard pooling with BlurPooling, utilizing Gaussian Error Linear Unit (GELU) activation, and adding self-attention after the Bi-LSTM layer, achieving an RMSE of 54.13 μm.

The modified VGG8 (BlurVGG8) was retained because it yielded better results, while ConvTran was used for the temporal component. To conduct an ablation study, three different APE variants were evaluated: Sin-APE, tAPE, and the proposed ldAPE (see Sect. 3.5). The ldAPE achieved the best performance, reaching an RMSE of 48.4μm (Table 2). To further assess the contribution of input velocity and the proposed BlurVGG8 architecture, we removed the velocity input and replaced BlurVGG8 with the original VGG8 in the best-performing configuration, observing the corresponding performance drop in each case.

Surface defects and their larger geometry create more complex time series patterns, increasing the error rate. One defect-free sample (I) and three samples with a block defect (800 μm thick) from the test set are visualized in Fig. 5a: sample II (1000 $\times$ 106 μm), sample III (2000 $\times$ 74 μm), and sample IV (3000 $\times$ 174 μm). In nearly all cases, the error rate decreased compared to the previous study that used predefined features. Specifically, the error changed from 30.8 μm to 33.9 μm for sample I, from 56.2 μm to 21.9 μm for sample II, from 50.4 μm to 49.3 μm for sample III, and from 82.8 μm to 57.1 μm for sample IV [12].

Table 1. Model comparison based on RMSE. Results are repeated over three independent runs for reliability.

Model Configuration	RMSE Avg.	RMSE std.
ViT + transformer encoder	148.2	3.6
VGG16 + BiLSTM	204.1	2.8
Pre-trained VGG16 + BiLSTM	81.9	9.1
VGG8 + BiLSTM	63.5	2.8
Pre-trained VGG8 + BiLSTM	86.1	4.0
BlurVGG8 + BiLSTM + Self attention	54.1	4.0
BlurVGG8 + ConvTran (ours)	**48.4**	2.4

Table 2. Ablation study on the effects of BlurVGG8, velocity input, and different APE variants.

Configuration	RMSE Avg.	RMSE std.
BlurVGG8 + ConvTran (Sin-APE)	53.8	6.1
BlurVGG8 + ConvTran (tAPE)	50.2	4.3
BlurVGG8 + ConvTran (ldAPE)	**48.4**	2.4
BlurVGG8 + ConvTran (ldAPE) without velocity	61.1	4.9
VGG8 + ConvTran (ldAPE)	57.5	4.1

Additionally, to evaluate the generalization capability, we compared their results on a sliding drop example that was not part of the training dataset. This experiment was performed on a hydrophobic surface (PFOTS_Si) with a block defect (800 μm thick, 3000 μm long, 23 μm high). While PFOTS_Si surfaces were in the training videos, this specific defect size was not. During the experiment, the drop stuck to the defect and detached very slowly, which had not occurred in the dataset. The model with predefined features based on domain knowledge produced an RMSE of 112.5 μm [12], while the model utilizing auto-extracted features achieved a significantly lower RMSE of 66.6 μm (Fig. 5b). We hypothesized that deep learning models with automated feature extraction would better capture complexities than those using predefined features. The RMSE improvement confirmed this. We altered the frames by adjusting illumination and introducing blurriness and artifacts, simulating challenging real-world conditions. The results indicated that the model's estimations remained robust under these perturbations, exhibiting minor fluctuations and slight deviations in the drop width measurements (Fig. 5c).

Feature Sensitivity. To evaluate how the model identifies key features for drop width estimation, we applied the Grad-CAM algorithm to visualize the Regions of Interest (ROIs) in the input images (Fig. 6). The figure presents seven middle frames from a sequence of 20, focusing on estimating the width of

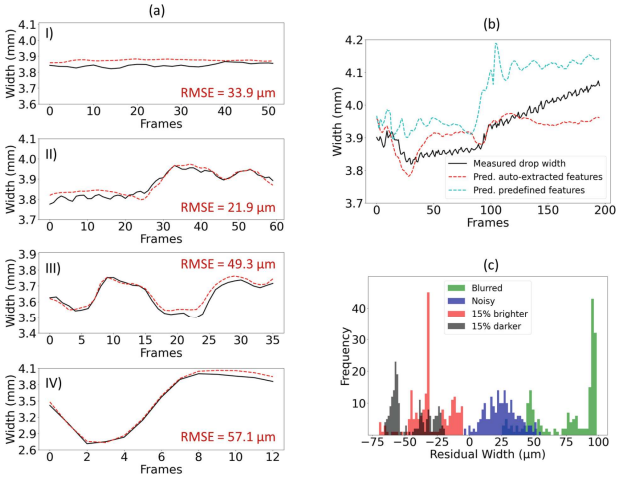


Fig. 5. a) Drop width measurements while sliding over a defect-free surface and three samples with defect, all 800 μm thick: sample II (length 1000 μm, height 106 μm), sample III (2000 $\times$ 74 μm), and sample IV (3000 $\times$ 174 μm). b) Comparison of the predefined features model and the automated feature extraction model on a sample outside the training dataset, with RMSEs of 112.5 μm and 66.6 μm, respectively. c) Effect of data augmentations (illumination changes, blurriness, and artifacts) on estimation diagrams using the automated feature extraction model. Distribution of residual errors (predicted - measured width) under different data augmentations. Each individual bar corresponds to the frequency of a specific residual value range.

the central frame (frame 10). Each row represents a different time step in the video sequence, illustrating how the model's attention dynamically shifts as the sliding drop interacts with the surface.

The sequence captures the critical moment when the advancing edge of the drop encounters a surface defect (Fig. 6, frame 7) and its subsequent response. The heatmaps in the middle column are spatially normalized between 0 and 1, ensuring that the most significant regions within each frame are distinctly highlighted. These visualizations reveal that the model consistently focuses on the drop's contour. The heatmaps in the left column remain unnormalized, preserving absolute activation values to capture spatiotemporal dependencies across frames. This enables a direct comparison of activation patterns over time. Notably, frames 8 and 9 exhibit the strongest activations, suggesting they provide the most critical information for accurately estimating the width of frame 10. This experiment demonstrates that the model effectively identifies key features aligned with established domain knowledge, such as drop length, height, and receding. Additionally, the Grad-CAM visualizations highlight the model's dynamic attention shifts, particularly at critical interaction points, reinforcing its ability to capture spatiotemporal dependencies. This opens the door for fur-

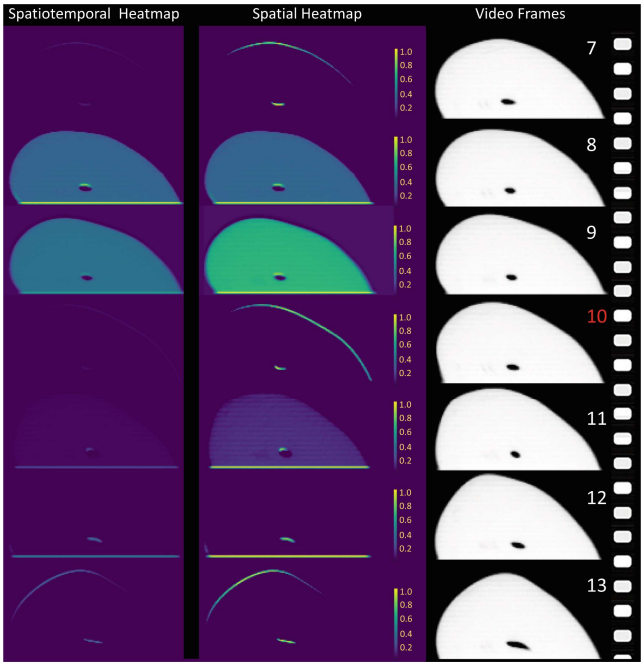


Fig. 6. Grad-CAM visualization of key regions influencing drop width estimation. The normalized heatmaps (middle column) emphasize critical spatial features, primarily the drop's edges, while the unnormalized heatmaps (left column) preserve absolute activation values, capturing spatiotemporal dependencies across frames.

ther studies to explore deeper feature correlations and refine automated methods for analyzing sliding drops.

5 Conclusions

In this study, we introduced a novel position invariant video processing method that effectively prevents overfitting to object location while reducing computation time by 84%. This is achieved by introducing the sliding spatiotemporal window (SSW) concept and incorporating the first derivative of the position of the region of interest into an architecture capable of processing both spatial and temporal data. The approach is scalable and can be extended to higher-dimensional cases, such as 2D object motion. Our approach, which leverages both a specialized VGG8-inspired architecture and the novel ldAPE representation, is well-suited for addressing spatiotemporal challenges with low information density, such as drop motion analysis. This method can effectively address challenges in drop and soft matter research. It is also applicable to scientific domains involving video sequences where the temporal contour evolution of small objects of interest is critical and data availability is limited, such as in biomedical video

analysis. Moreover, it integrates with interpretability techniques like Grad-CAM, offering deeper insights into model behavior by highlighting the most influential video features. The interpretability and performance of our method pave the way for uncovering new correlations. For example, we observed that subtle reflections within drops, although seemingly insignificant, may carry meaningful information about drop geometry. Our dataset includes variations in drop viscosity, surface chemistry, wettability, sliding angle, and surface defect geometry, enabling our research to address a broad range of physical conditions and support generalization. However, the current scope does not include phenomena such as slide electrification or extreme wetting regimes (e.g., superhydrophobic or superhydrophilic surfaces), which are left for future investigation. This approach is currently being applied in experimental workflows at the Max Planck Institute for Polymer Research to support automated drop analysis in surface science experiments. To support further research, we have made our code and dataset publicly available.

6 Supplementary Information

Several related works and additional implementation details are discussed in the Supplementary Information document, where the following references are also cited [12,33,44].

Acknowledgments. We thank Geoff Webb for the valuable scientific discussions. We acknowledge financial support from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (Grant Agreement No. 883631) (S.S., F.D., H.-J.B.). Additional support was provided by the Priority Programme 2171 Dynamic Wetting of Flexible, Adaptive, and Switchable Surfaces (Grant Nos. BU 1556/36 and BE 3286/6-1: H.-J.B., R.B.), and by the German Research Foundation (DFG) through the Collaborative Research Center (CRC) 1194 Interaction between Transport and Wetting Processes (Project-ID 265191195), project C07N and T02 (R.B., H.-J.B.).

Disclosure of Interests. The authors declare no competing interests.

References

1. Sbragaglia, M., et al.: Sliding drops across alternating hydrophobic and hydrophilic stripes. Phys. Rev. E **89**(1), 012406 (2014)
2. Yonemoto, Y., Suzuki, S., Uenomachi, S., Kunugi, T.: Sliding behaviour of water-ethanol mixture droplets on inclined low-surface-energy solid. Int. J. Heat Mass Transf. **1**(120), 1315–24 (2018)
3. Li, X., Bodziony, F., Yin, M., Marschall, H., Berger, R., Butt, H.J.: Kinetic drop friction. Nature Commun. **14**(1), 4571 (2023)
4. Extrand, C.W., Kumagai, Y.: Liquid drops on an inclined plane: the relation between contact angles, drop shape, and retentive force. J. Colloid Interface Sci. **170**(2), 515–21 (1995)

5. Furmidge, C.G.: Studies at phase interfaces. I. The sliding of liquid drops on solid surfaces and a theory for spray retention. J. Colloid Sci. **17**(4), 309–324 (1962)
6. Larkin, B.K.: Numerical solution of the equation of capillarity. J. Colloid Interface Sci. **23**(3), 305–12 (1967)
7. Frenkel, Y.I.: On the behavior of liquid drops on a solid surface. 1. The sliding of drops on an inclined surface. arXiv preprint physics/0503051 (2005)
8. Buzágh, A., Wolfram, E.: Bestimmung der Haftfähigkeit von Flüssigkeiten an festen Körpern mit der Abreißwinkelmethode. II. Kolloid-Zeitschrift. **157**, 50–3 (1958)
9. Extrand, C.W., Gent, A.N.: Retention of liquid drops by solid surfaces. J. Colloid Interface Sci. **138**(2), 431–42 (1990)
10. Gao, N., Geyer, F., Pilat, D.W., Wooh, S., Vollmer, D., Butt, H.J., Berger, R.: How drops start sliding over solid surfaces. Nat. Phys. **14**(2), 191–6 (2018)
11. Li, X., et al.: Spontaneous charging affects the motion of sliding drops. Nat. Phys. **18**(6), 713–9 (2022)
12. Shumaly, S., et al.: Estimating sliding drop width via side-view features using recurrent neural networks. Sci. Rep. **14**(1), 12033 (2024)
13. Yancheshme, A.A., Hassantabar, S., Maghsoudi, K., Keshavarzi, S., Jafari, R., Momen, G.: Integration of experimental analysis and machine learning to predict drop behavior on superhydrophobic surfaces. Chem. Eng. J. **1**(417), 127898 (2021)
14. Zhang, J., Lin, G., Yin, X., Zeng, J., Wen, S., Lan, Y.: Application of artificial neural network (ANN) and response surface methodology (RSM) for modeling and optimization of the contact angle of rice leaf surfaces. Acta Physiol. Plant. **42**, 1–5 (2020)
15. Kokalis, C.C., Tasakos, T., Kontargyri, V.T., Siolas, G., Gonos, I.F.: Hydrophobicity classification of composite insulators based on convolutional neural networks. Eng. Appl. Artif. Intell. **1**(91), 103613 (2020)
16. Roy, S.S., Paramane, A., Singh, J., Chatterjee, S.: Accurate hydrophobicity grade detection of polymeric insulators in extremely wetted and humid environments using Bi-LSTM neural network classifier. In: 2022 IEEE Power & Energy Society General Meeting (PESGM), pp. 1–5. IEEE (2022)
17. Rabbani, A., Sun, C., Babaei, M., Niasar, V.J., Armstrong, R.T., Mostaghimi, P.: DeepAngle: fast calculation of contact angles in tomography images using deep learning. Geoenergy Sci. Eng. **1**(227), 211807 (2023)
18. Kabir, H., Garg, N.: Machine learning enabled orthogonal camera goniometry for accurate and robust contact angle measurements. Sci. Rep. **13**(1), 1497 (2023)
19. Shumaly, S., et al.: Deep learning to analyze sliding drops. Langmuir **39**(3), 1111–22 (2023)
20. Ismail Fawaz, H., Forestier, G., Weber, J., Idoumghar, L., Muller, P.A.: Deep learning for time series classification: a review. Data Min. Knowl. Disc. **33**(4), 917–63 (2019)
21. Faouzi, J.: Time series classification: a review of algorithms and implementations. In: Machine Learning (Emerging Trends and Applications) (2022)
22. Lim, B., Zohren, S.: Time-series forecasting with deep learning: a survey. Phil. Trans. R. Soc. A **379**(2194), 20200209 (2021)
23. Torres, J.F., Hadjout, D., Sebaa, A., Martínez-Álvarez, F., Troncoso, A.: Deep learning for time series forecasting: a survey. Big data. **9**(1), 3–21 (2021)
24. Benidis, K., et al.: Deep learning for time series forecasting: tutorial and literature survey. ACM Comput. Surv. **55**(6), 1–36 (2022)
25. Tan, C.W., Bergmeir, C., Petitjean, F., Webb, G.I.: Time series extrinsic regression: predicting numeric values from time series data. Data Min. Knowl. Disc. **35**(3), 1032–60 (2021)

26. Sammut, C., Webb, G.I., (eds.) Encyclopedia of Machine Learning. Springer (2011)
27. Reiss, A., Indlekofer, I., Schmidt, P., Laerhoven, K.: Deep PPG: large-scale heart rate estimation with convolutional neural networks. Sensors **19**(14), 3079 (2019)
28. Zhang, Z., Pi, Z., Liu, B.: TROIKA: a general framework for heart rate monitoring using wrist-type photoplethysmographic signals during intensive physical exercise. IEEE Trans. Biomed. Eng. **62**(2), 522–31 (2014)
29. Dempster, A., Petitjean, F., Webb, G.I.: ROCKET: exceptionally fast and accurate time series classification using random convolutional kernels. Data Min. Knowl. Disc. **34**(5), 1454–95 (2020)
30. Ismail Fawaz, H., et al.: InceptionTime: finding alexnet for time series classification. Data Min. Knowl. Disc. **34**(6), 1936–62 (2020)
31. Mohammadi Farsani, R., Pazouki, E.: A transformer self-attention model for time series forecasting. J. Electr. Comput. Eng. Innovations (JECEI) **9**(1), 1 (2020)
32. Foumani, N.M., Tan, C.W., Webb, G.I., Salehi, M.: Improving position encoding of transformers for multivariate time series classification. Data Min. Knowl. Disc. **38**(1), 22–48 (2024)
33. Canny, J.: A computational approach to edge detection. IEEE Trans. Pattern Anal. Mach. Intell. **6**, 679–98 (1986)
34. Wen, Q., et al.: Transformers in time series: a survey. arXiv preprint arXiv:2202.07125 (2022)
35. Jiang, Z.P., Liu, Y.Y., Shao, Z.E., Huang, K.W.: An improved VGG16 model for pneumonia image classification. Appl. Sci. **11**(23), 11185 (2021)
36. Zhang, R.: Making convolutional networks shift-invariant again. In: International Conference on Machine Learning, pp. 7324–7334 (2019)
37. He, K., Chen, X., Xie, S., Li, Y., Dollár, P., Girshick, R.: Masked autoencoders are scalable vision learners. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 16000–16009 (2022)
38. Vaswani, A., et al.: Attention is all you need. In: Advances in Neural Information Processing Systems, p. 30 (2017)
39. Wu, K., Peng, H., Chen, M., Fu, J., Chao, H.: Rethinking and improving relative position encoding for vision transformer. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 10033–10041 (2021)
40. Dufter, P., Schmitt, M., Schütze, H.: Position information in transformers: an overview. Comput. Linguist. **48**(3), 733–63 (2022)
41. Boinovich, L.B., Emelyanenko, A.M.: Recent progress in understanding the anti-icing behavior of materials. Adv. Coll. Interface. Sci. **1**(323), 103057 (2024)
42. Ghasemlou, M., et al.: Self-lubricated, liquid-like omniphobic polymer brushes: advances and strategies for enhanced fluid and solid control. Prog. Polym. Sci. **19**, 101933 (2025)
43. Zhou, X., et al.: Thickness of nanoscale poly (Dimethylsiloxane) layers determines the motion of sliding water drops. Adv. Mater. **36**(29), 2311470 (2024)
44. Darvish, F., et al.: Control of spontaneous charging of sliding water drops by plasma-surface treatment. Sci. Rep. **14**(1), 10640 (2024)

5.2 Supporting information

Supplementary Information for "CNN-Transformer with absolute positional encoding optimized for low-dimensional inputs: Applied to estimate sliding drop width"

Sajjad Shumaly[1] (✉), Fahimeh Darvish[1], Mahsa Salehi[2], Navid Mohammadi Foumani[2], Oleksandra Kukharenko[1], Hans-Jürgen Butt[1], Ulrich Schwanecke[3], and Rüdiger Berger[1]

[1] Max Planck Institute for Polymer Research (MPI-P), Mainz, Germany
[2] Department of Data Science and Artificial Intelligence, Monash University, Melbourne, VIC, Australia
[3] Department of Computer Science and Media, RheinMain University of Applied Sciences, Wiesbaden, Germany

S.1 The sliding drop setup

The sliding drop setup includes a high-speed camera (FASTCAM Mini UX100, Photron) equipped with a TitanTL telecentric lens (×0.268, 1-inch, C-mount, Edmund Optics) to record the sliding drop videos (Figure S.1). Additionally, a telecentric backlight illuminator (Edmund Optics) is employed to ensure uniform illumination. The front view of the sliding drops is simultaneously captured using two parallel mirrors (25 × 36 mm^2 protected silver mirror; PFR10-P01, Thorlabs) placed on either side of the sample to reflect the backlight from the illuminator. All components are mounted on a breadboard that can be rotated from 0° to 90°, ensuring the alignment of the optical setup remains intact when the entire setup is rotated. Distilled water drops (<1 μS cm^{-1}; Gibco, Thermo Fisher Scientific) with a volume of 32 μL are deposited onto the top of a tilted plane using a peristaltic pump (MINIPULS 3, Gilson) connected to a grounded blunt syringe needle (1.5 mm outer diameter). By releasing the liquids from a height of approximately 5 mm, they detach from the syringe just before contacting the surface.

S.2 Sample preparation

Dataset Description We compiled a dataset comprising both defect-containing and defect-free samples to enable robust training and evaluation of our model. The dataset was designed to enhance the model's generalization capability by incorporating a diverse range of defect geometries and surface types.

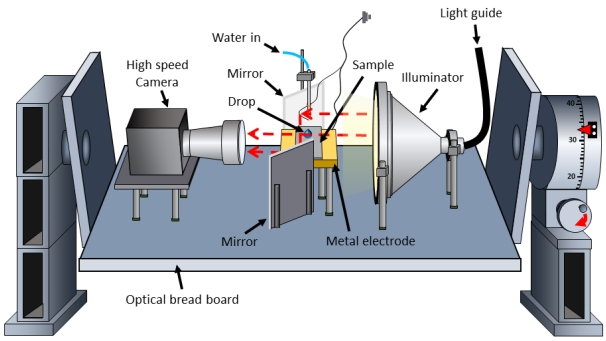


Fig. S.1. Experimental setup for recording videos of sliding drops. The drawing is adapted from [1].

Samples with Defects The samples with defects were fabricated by patterning SU-8 photoresist on silicon wafers. To induce variability in defect geometry, multiple types of structures were created, each differing in width, height, or shape. All samples with defects were coated with PFOTS (perfluorooctyltrichlorosilane).

- **Block defect-I:** thickness = 800 µm, width = 1000 µm, height = 106 µm
- **Block defect-II:** thickness = 800 µm, width = 2000 µm, height = 74 µm
- **Block defect-III:** thickness = 800 µm, width = 3000 µm, height = 174 µm
- **Block defect-IV (final validation sample):** thickness = 800 µm, width = 3000 µm, height = 23.0 µm
- **Cylindrical defect-I:** diameter = 800 µm, height = 31 µm
- **Cylindrical defect-II:** diameter = 800 µm, height = 47 µm
- **Step defect:** thickness = 800 µm, width = 2 cm, height = 30 µm

Samples without Defects Defect-free samples represent different material compositions and coatings. These include:

- **Thiols_Au:** A self-assembled monolayer of perfluorodecanethiol on gold-coated glass substrates.
- **PFOTS-Si:** Silicon wafers treated with 1H,1H,2H,2H-perfluorooctyltrichlorosilane.
- **Glycerol on Thiols_Au:** A series of samples where droplets containing 20%–40% glycerol concentrations were applied to Thiols_Au surfaces.
- **Miscellaneous:** Twelve additional recordings include surfaces such as PFOTS-coated SiO_2 and gold-coated glass, further enriching the dataset.

A total of 99 block defect, 33 cylindrical defect, and 15 step defect experiments were conducted, along with 88 experiments on defect-free samples. These experiments were performed under varying conditions, including changes in tilt angle, liquid viscosity, and surface chemistry. This diverse range of experiments aids in ensuring that the model does not overfit to specific chemistry and conditions [1].

The Thiols_Au sample was prepared using gold-coated glass substrates, created by sputter-coating a layer of 5 nm chromium followed by 35 nm gold (BalTec MED 020). A Teflon film, approximately 60 nm thick, was then coated onto the gold substrate via dip coating at a pulling speed of 10 mm/min from a solution of 1 wt% Teflon AF 1600 ($\varepsilon = 1.9$; Sigma-Aldrich) in FC-75 (97%, Fisher Scientific). The Teflon samples were ultimately annealed in a vacuum oven at 160°C for 24 hours before use.

To create samples with defects, silicon wafers were cleaned twice in acetone using ultrasonication, followed by 2-propanol. The cleaned Si wafers were exposed to O_2 plasma (Femto – Diener GmbH, Germany; 140 W, 5 min, 0.3 mbar). A filtered nitrogen gun was used to remove small particles from the Si wafers. Subsequently, 1 mL of SU-8 (GM1060, GM1070; Gersteltec Sàrl, Switzerland) was dropped onto the Si wafers. In GM1060, the spin speed was ramped from 500 rpm (7 s) to 1000 rpm (40 s) for block-defect samples. Spin speeds of 850 rpm (40 s) and 650 rpm (40 s) were used for cylindrical defects. In GM1070, the spin speed was ramped from 500 rpm (7 s) to 700 rpm, 1000 rpm, and 1500 rpm (40 s each) for various block defects. The samples were then soft-baked at 65 °C (30 min), 95 °C (2–4 min), and 65 °C (30 min), and subsequently cooled to room temperature. Afterwards, the samples were mounted on a mask aligner (MJB3, Süss MicroTec) for UV curing. Through a photomask, the SU-8 was exposed to UV light (8 s, 290 J/cm^2). Post-baking was performed at 95 °C for 1–2 min, then the samples were developed—i.e., the unexposed areas of SU-8 were dissolved in 1-methoxy-2-propanol acetate (CAS 108-65-6). The samples were rinsed in 2-propanol for 1 min and hard-baked (2 h, 150 °C).

The samples were hydrophobized by vapor-phase deposition of a fluorosilane (PFOTS, CAS 78560-45-9). PFOTS was selected for its low boiling point (180 °C), which enables gas-phase evaporation and enhances deposition reactivity. Prior to fluorination, the samples were activated with oxygen plasma (5 min, 140 W, 0.3 mbar). Fluorination was performed in a CVD chamber at a base pressure of 50–60 mbar. The reaction time was 30 min. Afterwards, to remove unbonded PFOTS, the samples were rinsed with ethanol and Milli-Q water, then post-evacuated (0.1 mbar) for 30 min [3].

S.3 Data augmentation

Data augmentation was employed to enhance the robustness and generalization capabilities of the model by simulating a variety of real-world conditions that might be encountered in practical applications. In operational environments, images of sliding drops may exhibit artifacts and variations due to environmental factors, or imperfections in imaging equipment (Figure S.2). Environmental

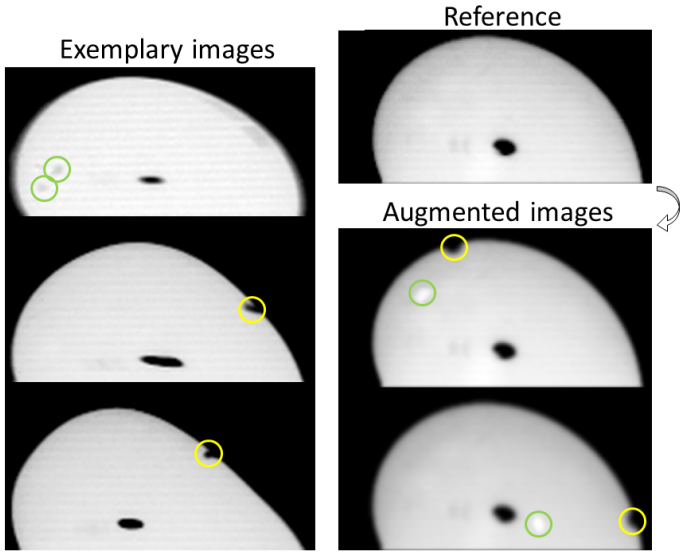


Fig. S.2. Illustration of data augmentations used to enhance the model's robustness and generalization capabilities. On the right side, a single reference image from the dataset is shown after applying augmentation techniques, including brightness adjustments, Gaussian blurring, and the introduction of synthetic image artifacts. On the left side, a sample of random images from the dataset in real-world operational settings is shown for comparison.

variations may include lighting fluctuations, dust within the lenses and scattered light that introduce noise in video frames, motion blur, and optical distortions like minor defocusing and focus inconsistencies.

The applied techniques included brightness adjustment, Gaussian blur filtering, and the addition of image artifacts. The brightness of the images was varied by up to $\pm 15\%$ to simulate lighting changes caused by ambient fluctuations or exposure variations. A Gaussian blur filter was applied using kernel sizes of 1×1 (no blurring), 3×3, and 5×5, randomly selected for each image. This simulated slight defocusing, motion blur, or focus imperfections. To mimic artifacts commonly encountered in imaging systems, two types of synthetic artifacts were introduced: irregular stains and radiance spots.

The irregular stains are artifacts simulate non-uniform patterns such as smudges or dirt on the imaging lens or sensor (Figure S.2, green circles). To generate irregular stains, we created random shapes by sampling points around a circle and applying sinusoidal perturbations to their radii, resulting in irreg-

ular contours. These shapes were then subjected to random transformations, including stretching, rotation, and scaling, to produce a variety of stain patterns. The stains were overlaid onto the images at random positions and with varying intensities. The related pseudo-code can be found in Algorithm 1.

Algorithm 1 Irregular Stains Augmentation

procedure StainAugmentation(img, max_size)
  Stain Size Computation:
  $stain\_size \leftarrow [0, max\_size]$, max_size = 0 indicating no spot.
  Stain Generation:
  Defines *angles* as evenly spaced points within $(0, 2\pi)$ along the circumference of a unit circle.
  Sample 5 random frequencies and phase shifts:

$$freq_i \in [0.5, 7], \phi_i \in [0, 2\pi]$$

  Adds sinusoidal perturbations to the unit circle, modifying its shape to create an irregular contour:

$$radius = 1.0 + \sum_{i=1}^{5} max\_deformation \times \sin(freq_i \times angles + \phi_i)$$

  Convert to Cartesian coordinates:

$$x = radius \cos(angles), \quad y = radius \sin(angles)$$

  Geometric Transformation:
  Randomly scale x or y by a factor in $[0.5, 3]$.
  Rotate by $\theta \in [0, \pi/2]$:

$$x_{rot} = x\cos\theta - y\sin\theta, \quad y_{rot} = x\sin\theta + y\cos\theta$$

  Stain Mask Creation:
  Create a binary mask and scale (x, y) to fit within $stain\_size$.
  Fill the corresponding polygon region in the mask.
  Gradient Filtering:
  Generate a circular gradient G of size $stain\_size$:

$$G(i,j) = 1 - \frac{\sqrt{(i-c)^2 + (j-c)^2}}{max\_distance}, \quad max\_distance = \sqrt{2} \times c, \quad c \equiv \text{stain center}$$

  Compute filtered gradient:

$$filtered\_gradient = G \times (1 - stain)$$

  Ensure non-zero values in $filtered\_gradient$.
  Stain Placement:
  Overlay the filtered gradient onto img inside of the drop.
  Output: return img
end procedure

We also simulated radiance spots, localized areas resembling lens flares or reflections. To generate these spots, we first applied the Canny edge detection algorithm [2] to identify drop edges in the images where such artifacts might naturally appear. Circular gradients were then created with intensity decreasing from the center outward to mimic light dispersion in radiance spots. The size, intensity, and position of the radiance spots were randomized for each image (Figure S.2, yellow circles), ensuring that the model is exposed to a wide range of such artifacts. The related pseudo-code can be found in Algorithm 2.

Algorithm 2 Radiance Spots Generation

procedure RADIANCESPOT(img, max_radius)
  Preprocessing:
  Invert the image contrast and convert to `uint8`
  Edge Detection:
  Apply an edge detection algorithm.
  Identify edge pixels and store their coordinates.
  Edge Selection:
  Randomly choose a coordinate (x, y) from the detected edge pixels.
  Glow Effect:
  Assign a random radius $r \in (0, max\_radius)$, $r = 0$ indicating no spot.
  Radiance Spot Generation:
  Initialize a gradient matrix with zeros.
  for each pixel (x', y') in the region $(x - r, y - r)$ to $(x + r, y + r)$ **do**
    Compute Euclidean distance:

$$d = \sqrt{(x' - x)^2 + (y' - y)^2}$$

    if $d \leq r$ **then**
      Set gradient intensity:

$$I(x', y') = 1 - \frac{d}{r}$$

    end if
  end for
  Blend the gradient with original intensities to simulate the glow effect.
  Output:
  Return the modified image with the applied glow effect.
end procedure

S.4 Low Dimension Absolute Position Encoding

A detailed analysis of ldAPE performance across different dimensions is illustrated in Figure S.3. The effectiveness of ldAPE is more pronounced in lower dimensions, where standard encoding techniques such as Sin-APE or tAPE show noticeable limitations in preserving positional information. This makes it a particularly effective choice for time-series applications, where lower-dimensional

feature representations are common. As dimensions increase other methods become more viable.

References

1. Shumaly, S., Darvish, F., Li, X., Kukharenko, O., Steffen, W., Guo, Y., Butt, H.-J., Berger, R.: Estimating sliding drop width via side-view features using recurrent neural networks. *Scientific Reports* **14**(1), 12033 (2024).
2. Canny, J.: A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (6), 679–698 (1986).
3. Darvish F, Shumaly S, Li X, Dong Y, Diaz D, Khani M, Vollmer D, Butt HJ. Control of spontaneous charging of sliding water drops by plasma-surface treatment. Scientific Reports. 2024 May 9;14(1):10640.

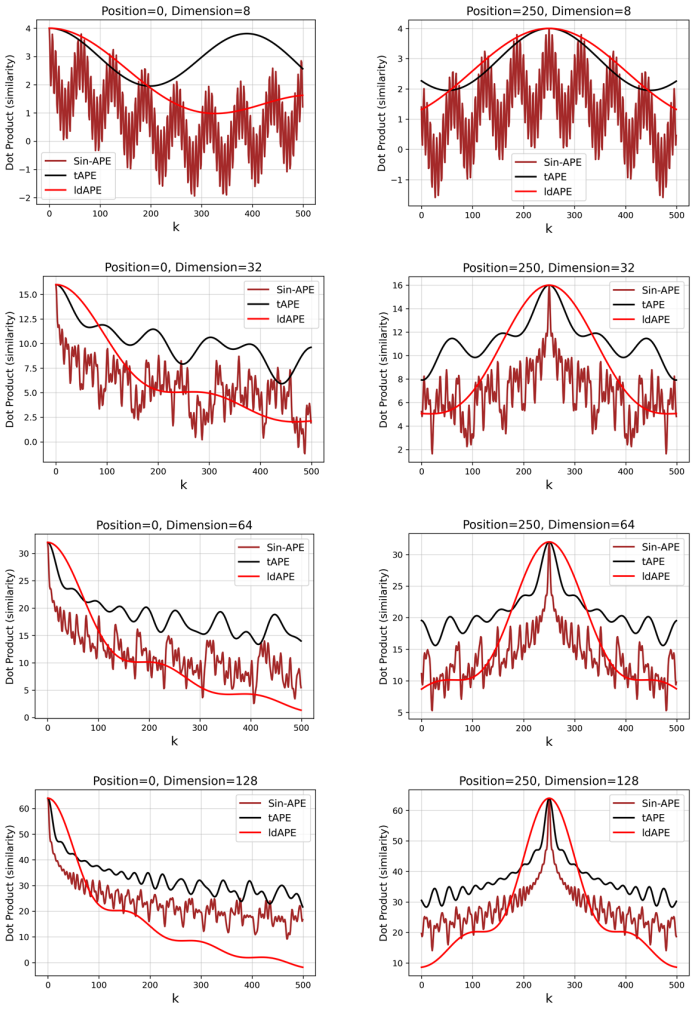


Fig. S.3. Performance comparison of different position encoding methods (ldAPE, Sin-APE, and tAPE) across various dimensions (8, 32, 64, and 128). The results demonstrate that ldAPE maintains a strong advantage in preserving positional information in lower dimensions, while Sin-APE and tAPE become more effective as the dimension increases.

Bibliography

[1] Diego Díaz, Aman Bhargava, Franziska Walz, Azadeh Sharifi, Sajjad Summaly, Rüdiger Berger, Michael Kappl, et al. Stood-up drop to determine receding contact angles. *Soft Matter*, 22(3):657–667, 2026. 1

[2] Fahimeh Darvish, Sajjad Shumaly, Xiaomei Li, Yun Dong, Diego Diaz, Mohammadreza Khani, Doris Vollmer, and Hans-Jürgen Butt. Control of spontaneous charging of sliding water drops by plasma-surface treatment. *Scientific Reports*, 14(1):10640, 2024. 1

[3] Fahimeh Darvish, Mark Isaacs, Sajjad Shumaly, Lea Delance, and Hans-Jürgen Butt. Water drops sliding over arrays of janus micropillars with hydrophilic tops: A new mechanism of drop charging. *Small*, page e11728, 2026. 1

[4] Xiaoteng Zhou, Pranav Sudersan, Diego Diaz, Benjamin Leibauer, Chirag Hinduja, Fahimeh Darvish, Pravash Bista, et al. Chemically robust superhydrophobic surfaces with a self-replenishing nanoscale liquid coating. *Droplet*, 3(1):e103, 2024. 1

[5] Malcolm E Schrader. Young-dupré revisited. *Langmuir*, 11(9):3585–3589, 1995. 2

[6] Hans-Jurgen Butt, Rudiger Berger, Werner Steffen, Doris Vollmer, and Stefan AL Weber. Adaptive wetting—adaptation in wetting. *Langmuir*, 34(38):11292–11304, 2018. 2

[7] Robert N Wenzel. Resistance of solid surfaces to wetting by water. *Industrial & engineering chemistry*, 28(8):988–994, 1936. 3

[8] David Quéré. Wetting and roughness. *Annu. Rev. Mater. Res.*, 38(1):71–99, 2008. 3

[9] ABD Cassie and SJToTFS Baxter. Wettability of porous surfaces. *Transactions of the Faraday society*, 40:546–551, 1944. 4

[10] RE Johnson. Wettability and contact angles. *Surface and colloid science*, 2:85, 1969. 4

[11] Nan Gao, Florian Geyer, Dominik W Pilat, Sanghyuk Wooh, Doris Vollmer, Hans-Jürgen Butt, and Rüdiger Berger. How drops start sliding over solid surfaces. *Nature Physics*, 14(2):191–196, 2018. 4, 9

[12] DW Pilat, P Papadopoulos, D Schaffel, Doris Vollmer, Rüdiger Berger, and H-J Butt. Dynamic measurement of the force required to move a liquid drop on a solid surface. *Langmuir*, 28(49):16812–16820, 2012. 4

[13] AM Gaudin, AF Witt, and TG Decker. Contact angle hysteresis–principles and application of measurement methods. *Trans*, pages 107–112, 1963. 4

[14] CO Timmons and WA Zisman. The effect of liquid structure on contact angle hysteresis. *Journal of Colloid and Interface Science*, 22(2):165–171, 1966. 4

[15] Nolwenn Le Grand, Adrian Daerr, and Laurent Limat. Shape and motion of drops sliding down an inclined plane. *Journal of Fluid Mechanics*, 541:293–315, 2005. 4

[16] Bekir Sami Yilbas, Abudllah Al-Sharafi, Haider Ali, and Nasser Al-Aqeeli. Dynamics of a water droplet on a hydrophobic inclined surface: influence of droplet size and surface inclination angle on droplet rolling. *Rsc Advances*, 7(77):48806–48818, 2017. 4

[17] Hans-Jürgen Butt, Karlheinz Graf, and Michael Kappl. *Physics and chemistry of interfaces*. John Wiley & Sons, 2023. 5

[18] Pierre-Gilles De Gennes, Françoise Brochard-Wyart, and David Quéré. *Capillarity and wetting phenomena: drops, bubbles, pearls, waves*. Springer Science & Business Media, 2003. 5

[19] CGL Furmidge. Studies at phase interfaces. i. the sliding of liquid drops on solid surfaces and a theory for spray retention. *Journal of colloid science*, 17(4):309–324, 1962. 5

[20] E Wolfram. Liquid drops on a tilted plate, contact angle hysteresis and the young contact angle. *Wetting, spreading and adhesion*, pages 213–222, 1978. 5

[21] RA Brown, FM Orr Jr, and LE Scriven. Static drop on an inclined plate: Analysis by the finite element method. *Journal of Colloid and Interface Science*, 73(1):76–87, 1980. 5

[22] Charles W Extrand and Y Kumagai. Liquid drops on an inclined plane: the relation between contact angles, drop shape, and retentive force. *Journal of colloid and interface science*, 170(2):515–521, 1995. 5

[23] Hans-Jürgen Butt, Jie Liu, Kaloian Koynov, Benedikt Straub, Chirag Hinduja, Ilia Roismann, Rüdiger Berger, Xiaomei Li, Doris Vollmer, Werner Steffen, et al. Contact angle hysteresis. *Current Opinion in Colloid & Interface Science*, 59:101574, 2022. 6, 7

[24] Ho-Young Kim, Heon Ju Lee, and Byung Ha Kang. Sliding of liquid drops down an inclined solid surface. *Journal of colloid and interface science*, 247(2):372–380, 2002. 7, 8

[25] Emmanuelle Rio, Adrian Daerr, Bruno Andreotti, and Laurent Limat. Boundary conditions in the vicinity of a dynamic contact line: Experimental investigation<? format?> of viscous drops sliding down an inclined plane. *Physical review letters*, 94(2):024503, 2005. 7

[26] Nis Korsgaard Andersen and Rafael Taboryski. Drop shape analysis for determination of dynamic contact angles by double sided elliptical fitting method. *Measurement Science and Technology*, 28(4):047003, 2017. 7, 10

[27] Francisco Bodziony, Xiaomei Li, Mariana Yin, Rüdiger Berger, Hans-Jürgen Butt, and Holger Marschall. Contribution of wedge and bulk viscous forces in droplets moving on inclined surfaces. *Theoretical and Computational Fluid Dynamics*, 38(4):583–601, 2024. 7

[28] Alice Pelosse, Élisabeth Guazzelli, and Matthieu Roché. Probing dissipation in spreading drops with granular suspensions. *Journal of Fluid Mechanics*, 955:A7, 2023. 8

[29] Andrea Montessori, Michele La Rocca, Pietro Prestininzi, Adriano Tiribocchi, and Sauro Succi. Deformation and breakup dynamics of droplets within a tapered channel. *Physics of Fluids*, 33(8), 2021. 8

[30] Terence D Blake and John M Haynes. Kinetics of liquidliquid displacement. *Journal of colloid and interface science*, 30(3):421–423, 1969. 8

[31] Jean-François Joanny and Pierre-Gilles De Gennes. A model for contact angle hysteresis. *The journal of chemical physics*, 81(1):552–562, 1984. 8

[32] Bruno Andreotti and Jacco H Snoeijer. Statics and dynamics of soft wetting. *Annual review of fluid mechanics*, 52(1):285–308, 2020. 8

[33] Xiaomei Li, Pravash Bista, Amy Z Stetten, Henning Bonart, Maximilian T Schür, Steffen Hardt, Francisco Bodziony, Holger Marschall, Alexander Saal, Xu Deng, et al. Spontaneous charging affects the motion of sliding drops. *Nature Physics*, 18(6):713–719, 2022. 8, 9

[34] Xiaoteng Zhou, Yongkang Wang, Xiaomei Li, Pranav Sudersan, Katrin Amann-Winkel, Kaloian Koynov, Yuki Nagata, Rüdiger Berger, and Hans-Jürgen Butt. Thickness of nanoscale poly (dimethylsiloxane) layers determines the motion of sliding water drops. *Advanced Materials*, 36(29):2311470, 2024. 8

[35] Ludmila B Boinovich and Alexandre M Emelyanenko. Recent progress in understanding the anti-icing behavior of materials. *Advances in Colloid and Interface Science*, 323:103057, 2024. 8

[36] Mehran Ghasemlou, Callum Stewart, Shima Jafarzadeh, Mina Dokouhaki, Motilal Mathesh, Minoo Naebe, and Colin J Barrow. Self-lubricated, liquid-like omniphobic polymer brushes: advances and strategies for enhanced fluid and solid control. *Progress in Polymer Science*, page 101933, 2025. 8

[37] OI Del Rıo and AW Neumann. Axisymmetric drop shape analysis: computational methods for the measurement of interfacial properties from the shape and dimensions of pendant and sessile drops. *Journal of colloid and interface science*, 196(2):136–147, 1997. 9

[38] Aurélien F Stalder, Tobias Melchior, Michael Müller, Daniel Sage, Thierry Blu, and Michael Unser. Low-bond axisymmetric drop shape analysis for surface tension and contact angle measurements of sessile drops. *Colloids and Surfaces A: Physicochemical and Engineering Aspects*, 364(1-3):72–81, 2010. 9

[39] CW Extrand. Uncertainty in contact angle measurements from the tangent method. *Journal of adhesion science and Technology*, 30(15):1597–1601, 2016. 10

[40] Darren L Williams, Anselm T Kuhn, Mark A Amann, Madison B Hausinger, Megan M Konarik, and Elizabeth I Nesselrode. Computerised measurement of contact angles. *Galvanotechnik*, 101(11):2502, 2010. 10

[41] CA Papakonstantinou, H Chen, and A Amirfazli. New ellipse-fitting method for contact angle measurement. *Surface Innovations*, 10(6):387–394, 2022. 10

[42] Aurélien F Stalder, Gerit Kulik, Daniel Sage, Laura Barbieri, and Patrik Hoffmann. A snake-based approach to accurate determination of both contact points and contact angles. *Colloids and surfaces A: physicochemical and engineering aspects*, 286(1-3):92–103, 2006. 11

[43] Emoke Albert, Borbala Tegze, Zoltan Hajnal, Daniel Zambo, Daniel P Szekrenyes, Andras Deak, Zoltan Horvolgyi, and Norbert Nagy. Robust contact angle determination for needle-in-drop type measurements. *ACS omega*, 4(19):18465–18471, 2019. 11

[44] Philip T Reiss, Jeff Goldsmith, Han Lin Shang, and R Todd Ogden. Methods for scalar-on-function regression. *International Statistical Review*, 85(2):228–249, 2017. 12

[45] Marco AF Pimentel, Peter H Charlton, and David A Clifton. Probabilistic estimation of respiratory rate from wearable sensors. In *Wearable Electronics Sensors: For Safe and Healthy Living*, pages 241–262. Springer, 2015. 12

[46] Chang Wei Tan, Christoph Bergmeir, François Petitjean, and Geoffrey I Webb. Time series extrinsic regression: Predicting numeric values from time series data. *Data Mining and Knowledge Discovery*, 35(3):1032–1060, 2021. 12

[47] George EP Box, Gwilym M Jenkins, Gregory C Reinsel, and Greta M Ljung. *Time series analysis: forecasting and control*. John Wiley & Sons, 2015. 12

[48] Jeffrey L Elman. Finding structure in time. *Cognitive science*, 14(2):179–211, 1990. 12, 13

[49] Tomas Mikolov, Martin Karafiát, Lukas Burget, Jan Cernockỳ, and Sanjeev Khudanpur. Recurrent neural network based language model. In *Interspeech*, volume 2, pages 1045–1048. Makuhari, 2010. 12, 13

[50] Ian Goodfellow, Yoshua Bengio, Aaron Courville, and Yoshua Bengio. *Deep learning*, volume 1. MIT press Cambridge, 2016. 13, 14, 16

[51] Alex Graves, Marcus Liwicki, Santiago Fernández, Roman Bertolami, Horst Bunke, and Jürgen Schmidhuber. A novel connectionist system for unconstrained handwriting recognition. *IEEE transactions on pattern analysis and machine intelligence*, 31(5):855–868, 2008. 13

[52] Herbert Jaeger and Harald Haas. Harnessing nonlinearity: Predicting chaotic systems and saving energy in wireless communication. *science*, 304(5667):78–80, 2004. 13

[53] Yoshua Bengio, Patrice Simard, and Paolo Frasconi. Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks*, 5(2):157–166, 1994. 13

[54] Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. On the difficulty of training recurrent neural networks. In *International conference on machine learning*, pages 1310–1318. Pmlr, 2013. 13

[55] Edmondo Trentin and Marco Gori. A survey of hybrid ann/hmm models for automatic speech recognition. *Neurocomputing*, 37(1-4):91–126, 2001. 13

[56] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997. 13

[57] Alex Graves, Abdel-rahman Mohamed, and Geoffrey Hinton. Speech recognition with deep recurrent neural networks. In *2013 IEEE international conference on acoustics, speech and signal processing*, pages 6645–6649. Ieee, 2013. 13

[58] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. *Advances in neural information processing systems*, 27, 2014. 13

[59] Jeffrey Donahue, Lisa Anne Hendricks, Sergio Guadarrama, Marcus Rohrbach, Subhashini Venugopalan, Kate Saenko, and Trevor Darrell. Long-term recurrent convolutional networks for visual recognition and description. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2625–2634, 2015. 13, 22

[60] Wojciech Zaremba, Ilya Sutskever, and Oriol Vinyals. Recurrent neural network regularization. *arXiv preprint arXiv:1409.2329*, 2014. 14

[61] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014. 14

[62] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. arxiv 2014. *arXiv preprint arXiv:1412.3555*, 1412, 2014. 14

[63] Duyu Tang, Bing Qin, and Ting Liu. Document modeling with gated recurrent neural network for sentiment classification. In *Proceedings of the 2015 conference on empirical methods in natural language processing*, pages 1422–1432, 2015. 14

[64] Yifan Guo, Weixian Liao, Qianlong Wang, Lixing Yu, Tianxi Ji, and Pan Li. Multidimensional time series anomaly detection: A gru-based gaussian mixture variational autoencoder approach. In *Asian conference on machine learning*, pages 97–112. PMLR, 2018. 14

[65] Rafal Jozefowicz, Wojciech Zaremba, and Ilya Sutskever. An empirical exploration of recurrent network architectures. In *International conference on machine learning*, pages 2342–2350. PMLR, 2015. 14

[66] Mike Schuster and Kuldip K. Paliwal. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45(11):2673–2681, 1997. 14

[67] Alex Graves and Jürgen Schmidhuber. Framewise phoneme classification with bidirectional LSTM and other neural network architectures. *Neural Networks*, 18(5–6):602–610, 2005. 16

[68] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 16

[69] George Zerveas, Srideepika Jayaraman, Dhaval Patel, Anuradha Bhamidipaty, and Carsten Eickhoff. A transformer-based framework for multivariate time series representation learning. In *Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining*, pages 2114–2124, 2021. 16

[70] Haoyi Zhou, Shanghang Zhang, Jieqi Peng, Shuai Zhang, Jianxin Li, Hui Xiong, and Wancai Zhang. Informer: Beyond efficient transformer for long sequence time-series forecasting. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 11106–11115, 2021. 16

[71] Bryan Lim, Sercan Ö Arık, Nicolas Loeff, and Tomas Pfister. Temporal fusion transformers for interpretable multi-horizon time series forecasting. *International journal of forecasting*, 37(4):1748–1764, 2021. 16

[72] Yuqi Nie, Nam H Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. A time series is worth 64 words: Long-term forecasting with transformers. *arXiv preprint arXiv:2211.14730*, 2022. 16

[73] Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and François Fleuret. Transformers are rnns: Fast autoregressive transformers with linear attention. In *International conference on machine learning*, pages 5156–5165. PMLR, 2020. 16

[74] Krzysztof Choromanski, Valerii Likhosherstov, David Dohan, Xingyou Song, Andreea Gane, Tamas Sarlos, Peter Hawkins, Jared Davis, Afroz Mohiuddin, Lukasz Kaiser, et al. Rethinking attention with performers. *arXiv preprint arXiv:2009.14794*, 2020. 16

[75] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 16, 18, 19

[76] Navid Mohammadi Foumani, Chang Wei Tan, Geoffrey I Webb, and Mahsa Salehi. Improving position encoding of transformers for multivariate time series classification. *Data mining and knowledge discovery*, 38(1):22–48, 2024. 17

[77] Lei Huang, Feng Mao, Kai Zhang, and Zhiheng Li. Spatial-temporal convolutional transformer network for multivariate time series forecasting. *Sensors*, 22(3):841, 2022. 17

[78] Qianqian Ren, Yang Li, and Yong Liu. Transformer-enhanced periodic temporal convolution network for long short-term traffic flow forecasting. *Expert Systems with Applications*, 227:120203, 2023. 17

[79] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 2002. 18

[80] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. pmlr, 2015. 18

[81] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012. 18

[82] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 18

[83] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015. 18

[84] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 18, 19

[85] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28, 2015. 18

[86] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *European conference on computer vision*, pages 21–37. Springer, 2016. 18

[87] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016. 18

[88] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European conference on computer vision*, pages 213–229. Springer, 2020. 19

[89] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10012–10022, 2021. 19

[90] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017. 19

[91] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *International conference on machine learning*, pages 10347–10357. PMLR, 2021. 19

[92] Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. Image super-resolution using deep convolutional networks. *IEEE transactions on pattern analysis and machine intelligence*, 38(2):295–307, 2015. 19

[93] Jiwon Kim, Jung Kwon Lee, and Kyoung Mu Lee. Accurate image super-resolution using very deep convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1646–1654, 2016. 19

[94] Christian Ledig, Lucas Theis, Ferenc Huszár, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, et al. Photo-realistic single image super-resolution using a generative adversarial network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4681–4690, 2017. 20

[95] Xintao Wang, Ke Yu, Shixiang Wu, Jinjin Gu, Yihao Liu, Chao Dong, Yu Qiao, and Chen Change Loy. Esrgan: Enhanced super-resolution generative adversarial networks. In *Proceedings of the European conference on computer vision (ECCV) workshops*, pages 0–0, 2018. 20

[96] Wenzhe Shi, Jose Caballero, Ferenc Huszár, Johannes Totz, Andrew P Aitken, Rob Bishop, Daniel Rueckert, and Zehan Wang. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1874–1883, 2016. 20

[97] Assaf Shocher, Nadav Cohen, and Michal Irani. "zero-shot" super-resolution using deep internal learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3118–3126, 2018. 20

[98] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3d convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pages 4489–4497, 2015. 22

[99] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. *Advances in neural information processing systems*, 27, 2014. 22

[100] Xingjian Shi, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-Kin Wong, and Wang-chun Woo. Convolutional lstm network: A machine learning approach for precipitation nowcasting. *Advances in neural information processing systems*, 28, 2015. 22

[101] Gedas Bertasius, Heng Wang, and Lorenzo Torresani. Is space-time attention all you need for video understanding? In *Icml*, volume 2, page 4, 2021. 23

[102] Anurag Arnab, Mostafa Dehghani, Georg Heigold, Chen Sun, Mario Lučić, and Cordelia Schmid. Vivit: A video vision transformer. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6836–6846, 2021. 23

[103] Sajjad Shumaly, Fahimeh Darvish, Xiaomei Li, Alexander Saal, Chirag Hinduja, Werner Steffen, Oleksandra Kukharenko, Hans-Jurgen Butt, and Rudiger Berger. Deep learning to analyze sliding drops. *Langmuir*, 39(3):1111–1122, 2023. 26

[104] Sajjad Shumaly, Fahimeh Darvish, Xiaomei Li, Oleksandra Kukharenko, Werner Steffen, Yanhui Guo, Hans-Jürgen Butt, and Rüdiger Berger. Estimating sliding drop width via side-view features using recurrent neural networks. *Scientific Reports*, 14(1):12033, 2024. 26

[105] Sajjad Shumaly, Fahimeh Darvish, Mahsa Salehi, Navid Mohammadi Foumani, Oleksandra Kukharenko, Hans-Jürgen Butt, Ulrich Schwanecke, and Rüdiger Berger. CNN-transformer with absolute positional encoding optimized for low-dimensional inputs: Applied to estimate sliding drop width. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 3–21, Cham, 2025. Springer Nature Switzerland. 27