MobiLyzer: Fine-grained Mobile Liquid Analyzer

SHAHRZAD MIRZAEI, Simon Fraser University, Canada
MARIAM BEBAWY, Simon Fraser University, Canada
AMR MOHAMED SHARAFELDIN, Simon Fraser University, Canada
MOHAMED HEFEEDA, Qatar Computing Research Institute, Qatar and Simon Fraser University, Canada

Most current methods for liquid analysis and fraud detection rely on expensive tools and controlled lab environments, making them inaccessible to lay users. We present MobiLyzer, a mobile system that enables fine-grained liquid analysis on unmodified commodity smartphones in realistic environments such as homes and grocery stores. MobiLyzer conducts spectral analysis of liquids based on how their chemical components reflect different wavelengths. Conducting spectral analysis of liquids on smartphones, however, is challenging due to the limited sensing capabilities of smartphones and the heterogeneity in their camera designs. This is further complicated by the uncontrolled nature of ambient illumination and the diversity in liquid containers. The ambient illumination, for example, introduces distortions in measured spectra, and liquid containers cause specular reflections that degrade accuracy. To address these challenges, MobiLyzer utilizes RGB images captured by regular smartphone cameras. It then introduces intrinsic decomposition ideas to mitigate the effects of illumination and interference from liquid containers. It further leverages the near-infrared (NIR) sensors on smartphones to collect complementary signals in the NIR spectral range, partially mitigating the limited sensing capabilities of smartphones. It finally presents a new machine-learning model that reconstructs the entire spectrum in the visible and NIR ranges using the captured RGB and NIR images, which enables fine-grained spectral analysis of liquids on smartphones without the need for expensive equipment. Unlike prior models, the presented spectral reconstruction model preserves the original RGB colors during reconstruction, which is critical for liquid analysis since many liquids differ only in subtle spectral cues. We demonstrate the accuracy and robustness of MobiLyzer through extensive experiments with multiple liquids, four different smartphones, and seven illumination sources. Our results show, for example, that MobiLyzer can accurately detect adulteration with small ratios, identify quality grades of the same liquid (e.g., refined vs. extra virgin olive oil), differentiate the country of origin of oils (e.g., olive oil from Italy versus USA), and analyze the concentration of materials in liquids (e.g., protein concentration in urine for early detection of kidney diseases).

CCS Concepts: • **Human-centered computing → Ubiquitous and mobile computing systems and tools**.

Additional Key Words and Phrases: ubiquitous computing, mobile sensing, smartphone imaging, spectral reconstruction, computer vision, liquid analysis, on-device AI

ACM Reference Format:
Shahrzad Mirzaei, Mariam Bebawy, Amr Mohamed Sharafeldin, and Mohamed Hefeeda. 2025. MobiLyzer: Fine-grained Mobile Liquid Analyzer. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 9, 4, Article 201 (December 2025), 37 pages. https://doi.org/10.1145/3770678

Authors' Contact Information: Shahrzad Mirzaei, Simon Fraser University, Burnaby, BC, Canada, shahrzad_mirzaei@sfu.ca; Mariam Bebawy, Simon Fraser University, Burnaby, BC, Canada, mariam_bebawy@sfu.ca; Amr Mohamed Sharafeldin, Simon Fraser University, Burnaby, BC, Canada, ams47@sfu.ca; Mohamed Hefeeda, Qatar Computing Research Institute, Doha, Qatar and Simon Fraser University, Burnaby, BC, Canada, mhefeeda@hbku.edu.qa.

ACM 2474-9567/2025/12-ART201
https://doi.org/10.1145/3770678

1 Introduction

Fraud in food and medical liquids poses significant health risks, erodes consumers' confidence, and incurs billions of dollars in costs each year. The scale of fraud is massive. For example, according to [11], approximately 34% of the inspected samples of expensive oils (e.g., almond, avocado, and flaxseed) did not meet the parameters declared on the labels. According to the US Food and Drug Administration (FDA) [28], there have been numerous instances of companies selling various oil mixtures or diluted olive oil as extra virgin oil. Further, studies from the World Health Organization (WHO) [52] indicate that many countries experience, on average, 10.5% of medicine adulteration. Recent investigations have uncovered the cough syrup scandal in Indonesia and India [53, 54], where these syrups contained toxic levels of ethylene glycol and diethylene glycol, resulting in kidney diseases in some children who consumed them. Another example is the adulteration of milk in India, where a recent study [25] shows the presence of various adulterants in milk, such as starch, neutralizers, and even detergents.

Current approaches for analyzing liquids and detecting cheating include high-performance liquid chromatography (HPLC) [57, 81], Raman spectroscopy [35, 61], thermal analysis [49], and metabolomics analysis [33, 57]. All of these approaches, however, require expensive equipment and laboratory settings.

Recent works, e.g., RF-EAST [74] and LiqRay [65], partially address this problem by analyzing the dielectric properties of liquids. However, they require special wireless devices, such as radio frequency identification (RFID) readers and ultra-wideband (UWB) transceivers, which are not readily available to lay users. Other works, e.g., LiquidHash [68], CapCam [80], Vi-Liquid [37], and Akte-Liquid [69], rely on smartphone cameras and microphones, which enable wide deployment. However, many of these approaches provide *coarse-grain* liquid analysis. That is, they can only analyze liquids with significantly different physical properties and detect adulteration when the adulterant ratios are high. Since many liquids have similar physical properties, such approaches may yield inaccurate results in realistic scenarios. For instance, mislabeling a lower-grade olive oil (e.g., refined) as extra virgin is a common form of fraud. Since both oil grades have very similar properties, e.g., viscosity, permittivity, and surface tension, prior approaches would likely result in low detection accuracy. Similarly, olive and peanut oils have similar viscosity [62]. Thus, approaches that rely on liquid viscosity would fail to detect adulteration of the olive oil with the much cheaper peanut oil.

In addition, multiple works make assumptions that are not easily met by lay users in everyday environments. For instance, CapCam [80], which measures surface tension using the phone camera, assumes that the liquid is transparent and the container bottom is visible from the surface. Akte-Liquid [69], which uses acoustic signals, requires placing the phone close to the liquid, necessitating opening the liquid bottles. Both approaches are susceptible to either illumination or audio interference from the surrounding environment, impacting their robustness and usability in real-world settings.

We present MobiLyzer, a system for *fine-grained* liquid analysis that works on unmodified smartphones in realistic environments, such as homes and grocery stores. Unlike prior works, MobiLyzer can accurately (i) detect adulteration with small ratios even if the adulterants have similar physical properties, (ii) distinguish quality grades of the same liquid (e.g., refined vs. extra virgin olive oil), (iii) differentiate the country of origin of oils (e.g., olive oil from Italy versus USA), and (iv) analyze material concentrations in liquids (e.g., protein concentration in urine for early detection of kidney diseases).

To perform fine-grained liquid analysis, MobiLyzer conducts accurate *spectral analysis* of liquids on smartphones, which captures how the chemical components of liquids interact with different wavelengths across the visible and near infrared (VNIR) range of the electromagnetic spectrum, which is between 400 and 1000 nm. However, conducting spectral analysis of liquids on smartphones is challenging due to the limited sensing capabilities of smartphones and the heterogeneity in their camera designs. For example, smartphone cameras only capture RGB (Red, Green, Blue) channels in the visible range, which is insufficient as several liquids have similar visual characteristics. This is further complicated by the uncontrolled nature of ambient illumination and

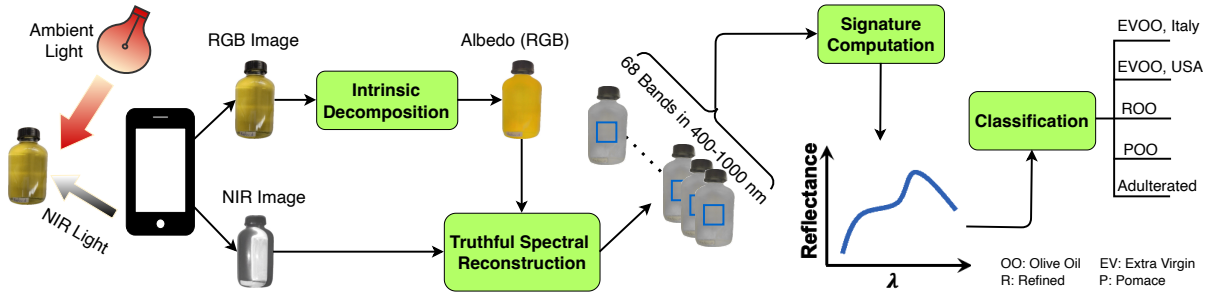

Fig. 1. Overview of MobiLyzer, which captures RGB and NIR images using unmodified phones. It mitigates the effects of container interference and ambient illumination by decomposing the RGB image and using only its essential component (albedo). It then upscales the albedo RGB and NIR images into N bands while preserving color truthfulness. It finally computes spectral signatures to analyze liquids. All models are designed and trained to generalize across phones.

the diversity in liquid containers. The ambient illumination introduces distortions in measured spectra, and liquid containers cause specular reflections that degrade accuracy.

To address these challenges, MobiLyzer utilizes RGB images captured by regular smartphone cameras. It then introduces intrinsic decomposition ideas to standardize data across different illuminations, thereby separating the illumination and container effects from the true colors of liquids. It further leverages the near-infrared (NIR) sensors on smartphones to collect complementary signals in the NIR spectral range, partially mitigating the limited sensing capabilities of smartphones. It finally presents a new machine-learning model that reconstructs the entire spectrum in the VNIR range using the captured RGB and NIR images, which enables fine-grained spectral analysis of liquids on smartphones without the need for expensive equipment. Unlike prior models, the presented spectral reconstruction model preserves the original RGB colors during reconstruction, which is crucial for liquid analysis, as many liquids differ only in subtle spectral cues.

In this paper, we address all of the above-mentioned research challenges and design MobiLyzer to be robust and general. Figure 1 presents an overview of MobiLyzer and its main components. The figure illustrates a simple example, where MobiLyzer can detect adulteration of olive oil and identify its quality grade and origin country. The source code of MobiLyzer and all datasets used in this paper are open source [50].

The contributions of this paper can be summarized as follows:

- We conduct an analysis using a hyperspectral camera to demonstrate the feasibility of identifying liquids using signals in the VNIR range (400–1000 nm), the same range available on phone cameras, in §3.1.
- We propose utilizing intrinsic decomposition to separate the true color of the liquid (diffuse albedo) from distortions caused by ambient illumination and reflections from the container, which leads to more accurate and consistent spectral analysis. This is presented in §4.2.
- We design a *truthful* spectral reconstruction (TSR) machine learning model, which takes RGB and NIR images captured by regular smartphones and reconstructs multiple wavelength bands across the entire VNIR range. In addition to achieving high reconstruction accuracy, TSR ensures color *truthfulness*—that is, the reconstructed spectra can accurately reproduce the original RGB and NIR images. This property preserves fine spectral cues essential for analyzing liquids. TSR is presented in §4.3.
- We analyze the essential aspects of smartphone heterogeneity and their impact on liquid analysis, and we present practical methods to handle such diversity, enabling the deployment of the proposed system across different phones. This is presented in §4.4.

- We implement a mobile application as a proof-of-concept on the Android platform, and we conduct an extensive study to analyze various aspects of different liquids, including detecting adulteration and grading of olive oil, determining fat and lactose percentages of milk, detecting adulteration of liquid medicine (Tylenol) and honey, and assessing protein content in urine for early detection of kidney diseases. Our results demonstrate the accuracy of MobiLyzer and its robustness to practical conditions. They also show that MobiLyzer outperforms recent works in the literature without requiring extra hardware or imposing unrealistic assumptions. Our evaluation is presented in §5.

2 Related Work

Liquid Analysis Using Wireless Signals. Several works measure various characteristics of wireless signals to analyze liquids, including LiquID [26], RF-EATS [74], LiqRay [65], PackquID [66], LiquImager [64], LiqDetector [76], FG-LiquID [43], and MetaBioLiq [20].

LiquID [26] measures time-of-flight, phase, and Received Signal Strength Indicator of Ultra-Wideband signals to model the liquid's electric permittivity. LiquID cannot process bottled liquids, and its accuracy is negatively affected by different container materials and shapes. RF-EATS [74] relies on coupling between RFID tags and receivers to analyze the dielectric properties of liquids inside their containers. Accurate functioning requires precise positioning of RFID tags and receivers, limiting its practical deployment, especially in dynamic environments. Moreover, it struggles to distinguish liquids with similar dielectric properties. LiqRay [65] uses a dual-antenna model to analyze liquid permittivity, adding hardware complexity and setup requirements.

PackquID [66] identifies liquids using RF signals embedded in packets, requiring specialized RF hardware not present in consumer smartphones. LiquImager [64] employs WiFi signals for fine-grained liquid identification and container imaging, but similarly relies on dedicated hardware and precise calibration. LiqDetector [76] leverages millimeter wave signals (mmWave) for robust container-independent liquid detection based on a dual-reflection model, necessitating specialized mmWave radar equipment. FG-LiquID [43] uses millimeter-wave sensing techniques for fine-grained liquid identification but depends on specialized external equipment. MetaBioLiq [20] utilizes wearable passive metasurface-aided mmWave sensing, providing high accuracy but relying on non-smartphone hardware.

These wireless-based methods require specialized hardware unavailable on consumer smartphones, significantly limiting their widespread practical adoption.

Liquid Analysis Using RGB Images. CapCam [80] analyzes surface tension through smartphone-captured wave patterns, requiring transparent liquids and visible bottom patterns, thus limiting general applicability. LiquidHash [68] relies on analyzing bubble characteristics, demanding transparent containers and specialized bottle caps, reducing practical deployment. Laser speckle-based methods [17] leverage smartphone LiDAR for viscosity measurement, yet sensitivity to ambient lighting and the required close proximity present limitations. The method in [36] detects adulteration using smartphone videos, but environmental illumination variations and narrow spectral range limit spectral analysis.

Current RGB image-based methods offer only coarse-grained liquid analysis. For instance, they are unable to detect small ratio liquid adulteration, as the subtle changes in surface tension and viscosity often fall below their detection threshold.

Liquid Analysis Using Acoustic and Other Signals. Vi-Liquid [37] measures viscosity through smartphone vibration motors and accelerometers but requires direct container contact and faces challenges with varying container shapes. Akte-Liquid [69] and Ls-liquid [70] measure acoustic impedance using smartphone speakers and microphones; however, environmental noise significantly affects accuracy. HearLiquid [79] also utilizes acoustic signals for fraud detection but similarly suffers from environmental audio interference. NutriLyzer [59] employs photo-acoustic sensing, necessitating specialized components, restricting its practical use.

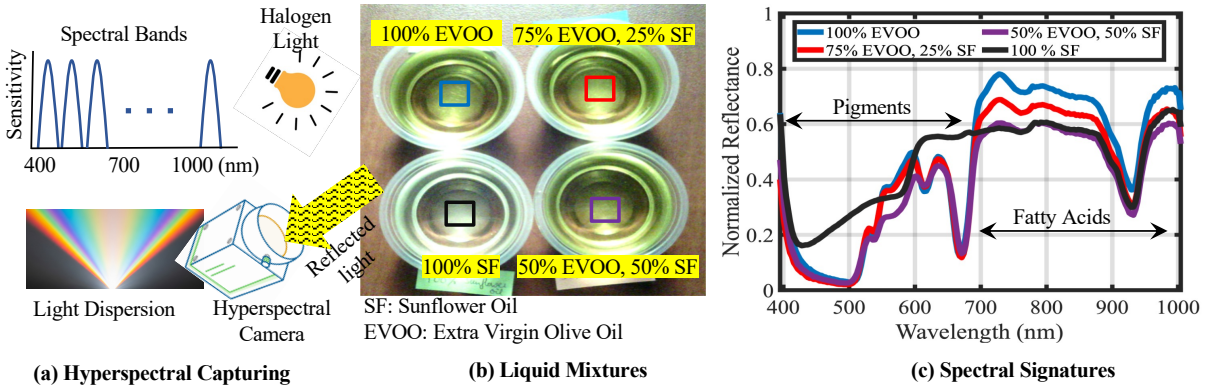

Fig. 2. Spectral analysis of olive and sunflower oil mixtures demonstrating the feasibility of differentiating among them. This is performed using an expensive hyperspectral camera under strict (halogen) illumination and open plastic containers.

Acoustic-based methods typically require direct liquid exposure, making them unsuitable for sealed containers and highly viscous liquids, such as honey and olive oil, where signal propagation is limited.

Our Work. MobiLyzer enables fine-grained spectral analysis using unmodified smartphone cameras, overcoming the need for expensive specialized hardware, direct liquid contact, constrained container shapes, and controlled environments. While prior work [67] focused on relatively matte surfaces, such as fruits, this work addresses the challenges of liquids in containers with complex light interactions by applying intrinsic decomposition to handle specularities and illumination variations and introducing a reconstruction model sensitive to subtle color differences critical for liquid analysis. We further demonstrate compatibility with various smartphone NIR sensing capabilities, supporting deployment across diverse devices.

3 Feasibility Study and Research Challenges

We first discuss the foundations of our solution and conduct a feasibility study using a hyperspectral camera to analyze liquids. Then, we summarize the research challenges to realize similar liquid analysis on smartphones.

3.1 Feasibility Study and Foundations of Liquid Spectral Analysis

Various materials absorb and reflect electromagnetic signals differently. Spectral analysis relies on this basic principle to differentiate among materials. A common approach for conducting spectral analysis is using expensive (tens of thousands of dollars) hyperspectral imaging systems [19]. Such systems capture the scene across many wavelengths (aka bands) using hyperspectral cameras, which use complex hardware such as light dispersion components and collimating lenses. The rich information captured by hyperspectral cameras allows the creation of *spectral signatures* of different materials, which are later used to identify them. A spectral signature is defined as the variation in intensity across wavelengths, and it is computed per pixel.

We provide a simplified illustration of hyperspectral imaging in Figure 2.a, where the reflected light from the scene is split into different bands, which are then recorded separately by the camera sensor. The sensitivity of the camera sensor is illustrated in the top-left part of the figure, indicating the fine-granularity of such cameras. The final output is a reflectance value in each spectral band and for each pixel. That is, the output is the three-dimensional image X, Y, Λ, where X, Y are the spatial resolution and Λ is the spectral resolution.

Commercial hyperspectral cameras operate in different ranges of the spectrum, including the visible and near-infrared (VNIR) (400–1000 nm) and short-wave infrared (SWIR) (1000–2500 nm). We focus on the VNIR

range because sensors in smartphone cameras operate in this range. Thus, we first need to establish the feasibility of conducting fine-grained liquid analysis using *only* the VNIR range, so that we could potentially do a similar analysis on phones that have much less capable cameras.

Specifically, we conduct experiments using a high-end hyperspectral camera (Model: Specim IQ, costs about 30K USD) that captures 204 bands in the 400–1000 nm range. Each band is a grayscale image with a spatial resolution of 512 × 512 pixels. We illuminate the scene using a halogen light source as recommended by the camera manufacturer. We analyze three different liquids and problems: (i) detecting adulteration in olive oil, (ii) identifying fat percentage in milk, and (iii) detecting kidney diseases via protein concentration in urine. We summarize the results for olive oil in this section and present the other two cases in §A.1.

Extra virgin olive oil (EVOO) is highly desirable because of its health benefits [48]. It is, however, significantly more expensive than other oils, such as sunflower (SF) oil, providing incentives for cheating. We prepare four different oil mixtures: (i) authentic 100% EVOO, (ii) 75% EVOO mixed with 25% SF, (iii) 50% EVOO mixed with 50% SF, and (iv) 100% SF. We put samples of the four oil mixtures in open plastic containers as shown in Figure 2.b. We compute a spectral signature for each oil mixture across the 204 bands captured by the camera. For robustness, and as typically done in hyperspectral applications [58], we average the signatures across the multiple pixels inside the 8x8 boxes marked in Figure 2.b.

We plot the signatures of different oil mixtures in Figure 2.c. The figure shows that the spectral signature of the 100% SF oil is quite different from that of the 100% EVOO, both in the visible range (due to the different colors as shown in the photos) and in the NIR range (due to the distinct fatty acid profiles). However, the differences among adulterated olive oils are very subtle. Since the colors do not change much for smaller percentages of added sunflower oil, the visible range may not provide enough information to distinguish the oil mixtures. The differences among signatures are more apparent in the NIR range. This is because NIR signals penetrate deeper beyond the oil surface and reveal more information about its chemical composition. Specifically, NIR signals are particularly effective for analyzing fat, sugars, and proteins in liquid samples due to the characteristic absorption patterns of their molecular bonds. Fat molecules with C-H bonds, sugars containing both C-H and O-H groups, and proteins featuring C-H, O-H, and N-H bonds all demonstrate identifiable overtone and combination vibrations within this spectral range [4, 55, 82].

3.2 Challenges of Conducting Spectral Analysis of Liquids on Smartphones

The previous section demonstrated that spectral analysis in the VNIR range can provide sufficient information to analyze various liquids. However, this analysis was conducted using expensive hyperspectral cameras in controlled laboratory settings (strict illumination, a single mounted camera, and the liquid samples were poured into open and transparent plastic containers). The goal of our work is to analyze liquids using diverse and unmodified smartphones operating in regular environments (e.g., homes and grocery stores) with arbitrary illumination, while the liquids can be in sealed containers. We summarize the main research challenges of achieving this goal in the following.

Challenge 1: Interference from Liquid Containers and Ambient Illumination. Prior works, e.g., [36, 67, 77], have analyzed solid foods and presented methods to mitigate the effects of diverse illumination sources. However, these methods cannot be applied to liquids. This is because, unlike fruits, liquids are typically kept in containers, which significantly interfere with how light interacts with liquids, compromising the accuracy of the spectral analysis. We illustrate this challenge in Figure 3. Figure 3.a shows light reflecting from a fruit, which has a mostly matte surface. The reflected signals from different depths can be captured to analyze the fruit. However, in the case of liquids, as Figure 3.b demonstrates, some signals are reflected from the container itself, which has nothing to do with the chemical composition of the liquid inside it. The container color may also partially mask the liquid color, which can lead to confusing different liquids. In addition, unlike laboratory settings, illumination in

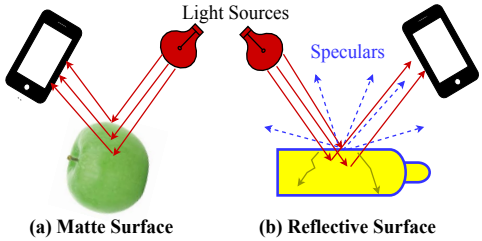

Fig. 3. Difficulty of analyzing liquids in containers.

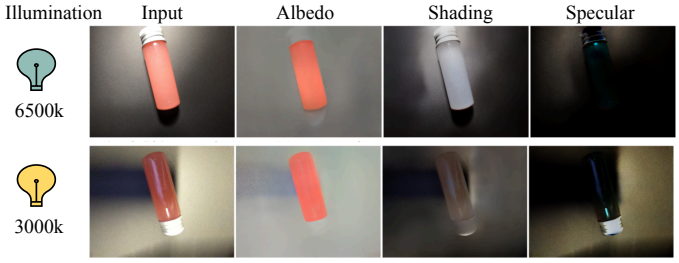

Fig. 4. Illustration of intrinsic decomposition in MobiLyzer.

everyday environments is arbitrary and comes from diverse sources, including LED bulbs with different color temperatures, various types of fluorescent bulbs, sunlight, and numerous mixtures of these and other sources.

The interference from containers and the diversity of illumination significantly damage the accuracy of spectral analysis, since this analysis relies on detecting tiny differences among spectral signatures, as discussed in §3.1. In §A.7, we conduct an experiment to demonstrate the impact of container interference on liquid signatures. To address this challenge, in §4.2, we separate the reflections from the container from those coming from the liquid inside it using *intrinsic decomposition* ideas. We also utilize intrinsic decomposition to mitigate the effect of illumination diversity. Intrinsic decomposition is a complex problem in computer vision [7, 23, 42]. While it has been proposed for various computer vision applications, such as image editing and relighting, it has not been considered before to facilitate spectral analysis of bottled liquids in environments with arbitrary illuminations, to the best of our knowledge. In addition, the solutions for intrinsic decomposition are computationally expensive. We introduce various optimizations that enable the execution of intrinsic decomposition on smartphones.

Challenge 2: Subtle Differences in Composition of Liquids. Adulterated liquids may have chemical compositions that do not differ significantly from authentic ones. For example, EVOO is chemically very similar to the lower-grade virgin olive oil (VOO). Additionally, mixing an expensive liquid with a small percentage of a cheaper liquid may not substantially alter the chemical composition of the resulting mixture. For example, the adulterated mixture of 75% EVOO and 25% SF has similar color and signature to the pure 100% EVOO, as shown in Figure 2.c. Therefore, accurate spectral signatures are required to conduct fine-grained liquid analysis for tasks such as fraud detection, quality differentiation, and concentration quantification. Such signatures need many spectral bands to create. However, smartphone cameras capture *only three bands*: Red (R), Green (G), and Blue (B), all of which fall within the visible range, as phones remove all signals beyond the visible range to avoid damaging the quality of RGB images. Three bands are insufficient to create spectral signatures, and the availability of NIR signals is crucial for analyzing liquid, as discussed in §3.1. In contrast, commercial hyperspectral cameras capture tens to hundreds of bands.

To address the problem of the limited number of captured bands by phones, we present, in §4.3, a *Truthful Spectral Reconstruction (TSR)* model that transforms pairs of RGB and NIR images captured by commodity smartphones into dense spectral bands across the entire VNIR (400–1000 nm) range. Unlike prior models, e.g., [10, 67], TSR preserves the true colors in the reconstructed bands. Liquid colors are important as they provide hints for differentiating liquids, unlike, for example, determining sugar contents in fruits, which relies on internal features. Minor color variations often signify chemical or physical differences in liquids. Without preserving these subtle visual and spectral nuances, the reconstructed bands could lead to unreliable analysis. In addition, we design and train TSR to be computationally efficient and deployable on various phones *without* requiring calibration or fine-tuning.

Table 1. Definitions of the main symbols used in the paper.

Symbol	Definition	Symbol	Definition
$I(\cdot)$	Image value	$R(\cdot)$	Surface reflectance
$L(\cdot)$	Illumination intensity	$R^a(\cdot)$	Diffuse albedo
$S(\cdot)$	Sensor sensitivity	$R^s(\cdot)$	Specular reflection
N	#Spectral bands	$\mathbf{I}, \mathbf{S}, \mathbf{R}$	Discrete matrix forms of $I(\cdot)$,S$(\cdot)$, $R(\cdot)$

Challenge 3: Heterogeneity of Smartphones. Modern smartphones exhibit considerable heterogeneity in hardware and software designs of their RGB and NIR camera systems, posing challenges for consistent and accurate spectral analysis and limiting deployability across different smartphones [2, 8, 22, 39]. For example, phone manufacturers may use various CMOS sensors in their RGB cameras. These sensors exhibit different sensitivity functions to the incident light, which means the resulting signals vary across phones, compromising the accuracy. In addition, manufacturers customize the image processing pipeline (ISP) on their phone cameras, which results in variations in the color distributions of images of the same scene captured by different phones, even when the illumination is the same. This, too, compromises the accuracy and limits the possibility of conducting accurate spectral analysis across phones. Furthermore, the NIR systems on phones are also diverse in terms of their structures and the wavelengths they utilize, making it more challenging to generalize the reconstruction model.

In §4.4, we examine the key aspects of phone diversity that impact spectral analysis. Then, we present practical solutions that enable accurate and general spectral analysis of liquids across phones.

4 Proposed Solution

4.1 Overview

As summarized in Figure 1, MobiLyzer functions as follows: A user takes RGB and NIR pictures of bottled liquids using an unmodified phone under regular illumination conditions. The system conducts spectral analysis of the captured liquid to perform the required task, such as detecting fraud in Olive oil, determining the type and fat percentage in milk, and assessing protein concentration in urine for early detection of kidney diseases.

MobiLyzer conducts spectral analysis of liquids in multiple steps. It first introduces a robust intrinsic image decomposition model, described in §4.2, to recover the true reflectance (albedo) of the liquid inside the bottle. This model is applied to the RGB image captured by the phone. It disentangles the albedo from specular reflections from the container and illumination-dependent artifacts. This enables the analysis of liquids in different types of containers without the need to open them.

MobiLyzer then utilizes the NIR image to provide much-needed information in the NIR range of the spectrum. However, both the RGB albedo and NIR images offer only four bands, which are still insufficient to produce spectral signatures such as those shown in Figure 2. In §4.3, we present a truthful spectral reconstruction model that takes the four (RGB albedo + NIR) bands and reconstructs many bands equally distributed in the entire VNIR range, which are similar to bands captured by real hyperspectral cameras. In §4.4, we describe various methods to make MobiLyzer deployable on different phones despite their heterogeneous designs. We present the design of our Android mobile application in §4.5, and we describe the limitations and extensions of MobiLyzer in §4.6.

4.2 Intrinsic Decomposition

Liquids in containers introduce significant optical complexities due to specular reflections, shadows, and varying illumination conditions. For example, as illustrated in Figure 3, specular reflections can overwhelm signals

reflected from the liquid inside the container, thereby masking subtle color differences that are critical for accurate analysis. We employ and customize intrinsic decomposing ideas to address these problems.

Definitions and Terminologies. Table 1 lists the symbols used in this paper. We denote by $I(x, y, n)$ the image value at pixel (x, y) and spectral band n which can be computed as [29]:

$$I(x, y, n) = \int S(n)\ L(x, y, \lambda)\ R(x, y, \lambda)\ d\lambda, \tag{1}$$

where $S(n)$ is the sensitivity function of the camera for the nth band, $L(x, y, \lambda)$ is the illumination intensity at position (x, y) which is a function of the wavelength λ, $R(x, y, \lambda)$ is the surface reflectance at position (x, y). For regular RGB cameras, n corresponds to one of the three color channels: R, G, and B. For hyperspectral cameras, n can be any of the N spectral bands. $S(\cdot)$ is specified by the sensor manufacturer, while $L(\cdot)$ depends on the ambient illumination. For the surface reflection $R(\cdot)$, light can be reflected in two primary ways: diffuse reflection and specular reflection. Diffuse reflection occurs when light is scattered equally in all directions from an object. Specular reflection happens when light reflects off a surface at the same angle at which it arrived, similar to a mirror [40, 63].

In the context of liquids inside sealed containers, diffuse reflection captures light that penetrates the container, interacts with the liquid, and reflects uniformly, thus encoding the liquid's intrinsic properties. This is referred to as ***diffuse albedo*** (or albedo for short), which we denote by $R^a(\cdot)$. Specular reflection, on the other hand, occurs predominantly from the surface of the container and is therefore not valuable for analyzing liquids. Specular reflection is denoted by $R^s(\cdot)$.

In controlled settings, it is possible to decompose the components of $R(\cdot)$ by minimizing specularities, estimating illumination, and calibrating the camera's spectral sensitivity. However, in real-world environments, this is difficult to achieve, making the intrinsic decomposition problem ill-posed and challenging to solve. [30]

Mitigating Effects of Containers and Illumination Using Intrinsic Decomposition. Conventional intrinsic decomposition methods, e.g., [31], approximate the surface reflectance as the product of $R^a(\cdot)$ and $R^s(\cdot)$. They make multiple simplifying assumptions, including uniform illumination and Lambertian surfaces that reflect light equally in all directions (and produce a matte appearance). Liquids within containers, however, exhibit non-Lambertian reflection characteristics. For example, liquid containers reflect light unevenly, producing complex light paths that vary significantly with illumination. Thus, the presence of containers complicates the disentanglement of the true liquid spectral characteristics from various artifacts, and conventional intrinsic decomposition methods would fail in this case.

For liquid analysis, we propose employing the recent data-driven intrinsic decomposition model in [13], which explicitly accounts for colorful shading and specular reflection. This model represents the image as: $I(\cdot) = R^a(\cdot)\ R^s(\cdot) + R^{res}(\cdot)$. The diffuse albedo $R^a(\cdot)$ is the true inherent color of the liquid itself. $R^s(\cdot)$ represents the colorful diffuse shading, capturing the effects of potentially multiple light sources as they interact diffusely with the scene. The term $R^{res}(\cdot)$ is defined as the residual layer, which explicitly captures non-diffuse illumination effects, including specular reflection and other complex scattering events that do not fit the diffuse reflection model.

Due to the complexity of the intrinsic decomposition problem in real-world settings, the solution is structured into three stages that gradually remove the single-color illumination and Lambertian world assumptions [13]. Each stage is a deep neural network. We introduce the subscripts g and c to denote gray and color shadings (or illuminations), respectively. For instance, $R^a_g(\cdot)$ and $R^a_c(\cdot)$ are the grayscale albedo and color albedo, respectively, both are computed during intermediary steps towards reaching the final diffuse albedo $R^a(\cdot)$. The first stage, ***Shading Chroma Estimation***, replaces the grayscale shading assumption with an RGB intrinsic diffuse model. A chroma network takes the grayscale shading $R^s_g(\cdot)$ and albedo $R^a_g(\cdot)$ from an off-the-shelf method [12], along with the input image, to estimate per-pixel shading chromaticity at low resolution. This chromaticity is then used

to construct a colorized RGB shading $R_c^s(\cdot)$, which helps prevent color shifts from being incorrectly embedded in the albedo. The second stage, ***Albedo Estimation***, uses the colorized shading $R_c^s(\cdot)$ from the previous stage to estimate a high-resolution, accurate diffuse albedo $R^a(\cdot)$. It leverages the sparse nature of albedo to remove residual illumination artifacts and color shifts, producing a flat albedo free from illumination effects. The final stage is the ***Diffuse Shading Estimation*** removes the Lambertian-world assumption. It uses a diffuse shading network that takes the refined diffuse albedo $R^a(\cdot)$, the colorized shading $R_c^s(\cdot)$, and the input image as a concatenated 9-channel input. It then computes the residual layer by subtracting the diffuse image $R^a(\cdot)$ $R_c^s(\cdot)$ from the input image, allowing for the representation of effects like specularities and over-exposed regions.

The recovered diffuse albedo, $R^a(\cdot)$, is the component that is fed into our subsequent spectral reconstruction model, presented in §4.3. This ensures that the generated spectral signature accurately reflects the true, intrinsic color properties of the liquid, free from the influence of arbitrary illumination and artifacts from the container.

Illustrative Example. In Figure 4, we present an example of our implementation of intrinsic decomposition to demonstrate its role in the analysis of liquids in sealed containers. The liquid in this example is Tylenol medicine, and the images were taken under two different LED illuminations. We modified the container directions to display various specularities and shadings. The figure shows the effectiveness of intrinsic decomposition in mitigating the effects of various illuminations and reflections from containers. Specifically, the albedo image appears clear and consistent despite variations in illumination and specular reflections.

Optimizations to Enable Intrinsic Decomposition on Smartphones. Intrinsic decomposition is a fundamental but very complex problem in computer vision. Our optimizations, which enable intrinsic decomposition in realistic environments on smartphones, can benefit many mobile applications, in addition to the liquid analysis problem considered in this paper.

We started from the open-source implementation [14] of the intrinsic decomposition model proposed by Careaga et al. [13] and developed a mobile-compatible adaptation that retains its core theoretical principles while addressing the practical constraints of mobile deployment, as also discussed in [71]. The original pipeline, while effective on desktop systems, was unsuitable for mobile platforms due to its dependence on desktop-specific Python libraries and its fragmented structure, which lacked the coherence required for efficient on-device execution. To resolve these issues, we restructured the pipeline into a unified, modular framework built on a tensor-centric architecture. Specifically, we reformulated the preprocessing and postprocessing stages using PyTorch [56], replacing external dependencies with functionally equivalent, PyTorch-native tensor operations. This redesign facilitates seamless export via the ONNX format [24], ensuring compatibility with a wide range of mobile inference engines and enabling efficient execution on resource-constrained devices.

Next, we applied dynamic quantization to reduce model size and inference time. Using this, we achieved a substantial reduction in memory usage (from 1.3 GB to 331 MB) and inference time (by up to 50%), balancing computational efficiency with minimal degradation in prediction accuracy. In §A.4, we provide more details on our optimizations. We also present experiments to: (i) validate that our optimizations did not significantly affect the accuracy of the intrinsic decomposition model, (ii) analyze various optimization trade-offs, and (iii) demonstrate that our optimizations enable the complex intrinsic decomposition model to run on smartphones with limited resources in real time.

4.3 TSR: Truthful Spectral Reconstruction in the VNIR Range

Need for Spectral Reconstruction. Spectral analysis involves creating spectral signatures to distinguish different materials, which requires many wavelength bands as illustrated in Figure 2.c. Hyperspectral cameras use complex optical, electronic, and mechanical components to split the reflected light from a scene into bands and then record these bands separately. Each band is a 2-D grayscale image. These bands provide the necessary information to compute accurate signatures. In contrast, RGB cameras capture only 3 bands, which are clearly

Fig. 5. Limitations of the state-of-the-art reconstruction model: It may yield inaccurate colors, affecting the accuracy of liquid analysis. Figure shows image pairs of original RGB (left) and reconstructed RGB (right).

too few to create representative spectral signatures. They also provide no information in the NIR range, which is essential to differentiate among liquids that have similar visual colors but different chemical compositions. To enable the creation of spectral signatures, we propose reconstructing multiple spectral bands from images captured by regular phones.

Limitations of Current Reconstruction Methods. Spectral reconstruction is a known problem in the computer vision community [6], with various applications including color enhancement and restoration. The state-of-the-art solution is the MST++ model [10], which is designed using vision transformers [27]. However, MST++ reconstructs bands *only* in the visible (400–700 nm) range. Recently, MobiSpectral [67] extended MST++ to take the NIR band captured by phones as an extra input to improve the reconstruction quality in the NIR range.

We first examine the suitability of MobiSpectral for liquid analysis using its open-source code and datasets. We trained their reconstruction model to produce 68 bands, using the same parameters specified in the paper. We computed the same accuracy metrics, including SAM (Spectral Angular Matching), SID (Spectral Information Divergence), and PSNR (Peak Signal-to-Noise Ratio). Our experiments confirm the results reported in [67]. These metrics, however, mostly assess the accuracy across the spectral domain but do not consider the *color truthfulness* aspect of the results. Color truthfulness of the reconstructed bands, which is also known as physical plausibility [44], is defined as the ability to obtain the original RGB image from the reconstructed bands.

To assess the color truthfulness of the reconstructed bands in the visible range, we perform the inverse operation of spectral reconstruction: we compute RGB images from the reconstructed bands. Then, we compare the original RGB images with the reproduced ones; ideally, they should be identical. We present sample results in Figure 5, which show significant color differences between the original and reproduced RGB images. Therefore, while MobiSpectral produces good accuracy across the spectral domain, it *does not guarantee* truthful reproduction of the original RGB colors, which is critical for analyzing liquids. This is because many liquids have subtle color differences, and losing such differences during the reconstruction process could compromise the accuracy of the spectral analysis. We note that studying liquids requires more detailed spectral analysis in both the visible and NIR ranges than the identification of organic fruits problem considered in [67], which mostly relies on the NIR range to examine the internal structure of fruits.

Proposed Truthful Spectral Reconstruction (TSR) Model. We propose a truthful spectral reconstruction model that takes as input the ***RGB (albedo)*** and NIR images and produces N bands equally distributed across the entire VNIR range. The RGB albedo image is computed by the intrinsic decomposition method described in §4.2, which isolates the effects of illumination and interference from the container. TSR reconstructs accurate bands similar to those captured by real hyperspectral cameras, as our evaluation in §5.1 shows. In addition, TSR ensures the truthfulness of both the RGB and NIR images; that is, the original RGB and NIR images can be computed from the reconstructed bands. As detailed below, TSR has two components. The first one *mathematically computes* four bands of the output N bands. The second component is a deep neural network that estimates the remaining $N - 4$ bands.

We next present the mathematical foundations of TSR. We rewrite Equation (1) in the discrete matrix form, while combining the illumination and reflectance spectrum together, as: $\mathbf{I} = \mathbf{S}^{\top}\ \mathbf{R}$, where $\mathbf{I}$ is a vector representing

the output image with four channels (R, G, B, and NIR), and $\mathbf{S}$ and $\mathbf{R}$ are matrices of dimensions $N \times 4$ for each pixel, where N is the number of spectral bands. $\mathbf{S}$ and $\mathbf{R}$ represent the sensor sensitivity and the reflectance spectrum across all bands, respectively. The spectral reconstruction problem defines a mapping function Φ to estimate/recover the entire reflectance spectrum from the 4-band input, that is, $\mathbf{R} = \Phi(\mathbf{I})$. The reconstruction is referred to as truthful if the estimated spectrum reproduces the same image, which can be ensured by:

$$\mathbf{I} = \mathbf{S}^\top \mathbf{R} = \mathbf{S}^\top \Phi(\mathbf{I}). \tag{2}$$

Thus, the goal is to design Φ such that it satisfies the constraint in Equation (2). To achieve this goal, and following [44], we divide the spectrum into two disjoint parts: $\mathbf{R} = \mathbf{R}^f + \mathbf{R}^b$, where $\mathbf{R}^f$ contains parts that contribute to the formation of the four RGB and NIR bands, and it is referred to as the fundamental metamer [21]. $\mathbf{R}^b$ contains the other $N-4$ spectral bands and is referred to as the metameric black.

$\mathbf{R}^f$ and $\mathbf{R}^b$ can be determined by projecting the reflectance spectrum $\mathbf{R}$ on $\mathbf{S}$ and $\mathbf{B}$, respectively, where $\mathbf{B}$ is an orthonormal matrix with dimensions $N \times (N-4)$ such that $\mathbf{B}^\top \mathbf{S} = 0$. Specifically, two orthogonal projection matrices are defined as [44]: $\mathbf{P}^S = \mathbf{S}(\mathbf{S}^\top \mathbf{S})^{-1}\mathbf{S}^\top$ and $\mathbf{P}^B = \mathbf{I} - \mathbf{P}^S$. $\mathbf{P}^S$ projects onto the column space of $\mathbf{S}$, and $\mathbf{P}^B$ projects onto the null space of $\mathbf{S}$. The fundamental metamer, which is the first component of TSR, is then computed as:

$$\mathbf{R}^f = \mathbf{P}^{\mathbf{S}}\mathbf{R} = \mathbf{S}(\mathbf{S}^\top \mathbf{S})^{-1}\, \mathbf{S}^\top \mathbf{R} = \mathbf{S}(\mathbf{S}^\top \mathbf{S})^{-1}\, \mathbf{I}. \tag{3}$$

On the other hand, the metameric black $\mathbf{R}^b$, which captures the spectral information that lies in the null space of the sensitivity matrix $\mathbf{S}$. It is computed by projecting the full reflectance spectrum $\mathbf{R}$ onto the space orthogonal to that of $\mathbf{S}$. The orthonormal basis $\mathbf{B}$ spans this space with dimensions $N \times (N-4)$, which satisfies: $\mathbf{B}^\top \mathbf{S} = \mathbf{0}$. Accordingly, the metameric black component is given by:

$$\mathbf{R}^b = \mathbf{P}^{\mathbf{B}}\mathbf{R} = (\mathbf{I} - \mathbf{P}^S)\mathbf{R} = \mathbf{B}\mathbf{b}, \tag{4}$$

where $\mathbf{b}$ is a coefficient vector of dimensions $(N-4) \times 1$ for each pixel. To obtain the orthonormal basis matrix $\mathbf{B} \in \mathbb{R}^{N \times (N-4)}$ for the null space of matrix $\mathbf{S}^\top \in \mathbb{R}^{4 \times N}$, we compute its *full* singular value decomposition (SVD): $\mathbf{S}^\top = \mathbf{U}\Sigma\mathbf{V}^\top$, where $\mathbf{U} \in \mathbb{R}^{N \times N}$ and $\mathbf{V} \in \mathbb{R}^{4 \times 4}$ are orthogonal matrices, and $\Sigma \in \mathbb{R}^{N \times 4}$ is a diagonal matrix containing the singular values. $\mathbf{B}$ is then computed as $\mathbf{B} = \mathbf{U}_{:,\,4+1:N}$. That is, it is formed by the last $(N-4)$ columns of $\mathbf{U}$. These columns span the null space of $\mathbf{S}^\top$.

The second component of TSR reconstructs $N-4$ spectral bands and is given by Equation (4), which requires computing the vector $\mathbf{b}$. We propose a learning-based approach to estimate $\mathbf{b}$ due to its complexity. Specifically, we design a neural network model to estimate $\mathbf{b}$. We base this model on the reconstruction model in [67], which extends MST++ [10]. We introduce new losses to enhance reconstruction accuracy, and we simplify the architecture, which substantially reduces its inference time (by up to 69%). The details are presented in §5.1.

Summary of TSR and its Training. The proposed TSR model ensures color truthfulness by extending the mathematical foundations in [44] to the entire VNIR range. TSR has two components. One computes four spectral bands using Equation (3). The other computes the remaining $N-4$ bands using Equation (4), which utilizes a neural network model. This model enhances the accuracy of the state-of-the-art reconstruction model by introducing new losses and simplifying its design to reduce computational complexity. The neural network in TSR requires training in a manner that allows deploying the trained model on different phones for inference. This, however, is not straightforward due to the diversity of phones and the model's dependence on phone characteristics. In §4.4, we present practical ideas for training the model to support its deployment on heterogeneous phones.

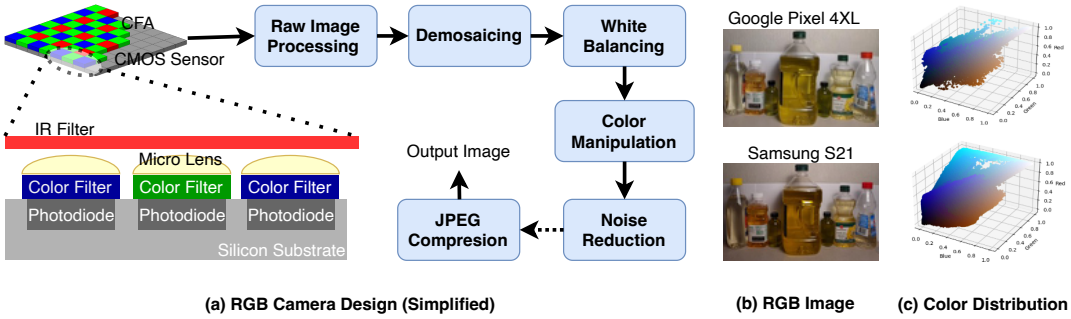

Fig. 6. Design of phone RGB cameras and the impact of their diversity on the produced images. The scene in (c) is captured under the same illumination but results in different colors in (d), which complicates spectral analysis across phones.

4.4 Supporting Liquid Analysis across Heterogeneous Smartphones

To make MobiLyzer deployable on a wide variety of smartphones, we analyze the key aspects of phone heterogeneity and their impact on the spectral analysis of liquids. We divide our analysis into two parts: RGB Camera Systems and NIR Camera Systems. In each case, we present solutions to handle phone heterogeneity.

RGB Camera Systems. We provide an overview of the design of smartphone RGB cameras in Figure 6.a. The primary hardware element is a CMOS sensor, which consists of a 2-dimensional array of photodiodes that convert light photons into electrical signals. A color filter array (CFA) covers the sensor, where each filter passes one of the RGB colors to the photodiode it covers. Microlenses are used to focus and direct photons toward photodiodes. While the sensor has sensitivity across the entire VNIR range, an infrared (IR) filter is typically used to remove all signals beyond the visible range (> 700 nm) to avoid damaging the visual quality. The electrical signals produced by the photodiodes are then processed through multiple steps to produce the RGB image. The second critical element in the design of RGB cameras is the image processing pipeline (ISP) shown in the figure, which includes multiple steps such as demosaicing, white balancing, and color manipulation.

The RGB camera system poses two problems for the spectral analysis of liquids. The first is the variation in the sensitivity of the CMOS sensor $\mathbf{S}$. Although the differences in sensitivity between various sensors may appear small, their impact is significant. This is because the sensor sensitivity directly affects the amount of electrical current generated by each photodiode, which is then fed and processed through the subsequent steps. That is, the differences are propagated and amplified through the image processing pipeline.

The second problem is that most of the processing pipeline in Figure 6.a is proprietary, as phone manufacturers compete on image quality. This means objects captured by various phones may look different *even under the same illumination*. We present an example in Figure 6 to demonstrate this challenge, where we captured the same scene in Figure 6.b under the same settings (e.g., illumination and distance) using two different phones: Google Pixel 4XL and Samsung S21. We plot the color distributions of the produced images in Figure 6.c, which show notable differences in the colors. The differences become more pronounced under diverse illuminations. Since spectral analysis relies on how pixel values change across different wavelengths, inconsistent pixel values from different phones will limit the possibility of conducting accurate spectral analysis across phones.

In our initial experiments, training our models on RGB images captured by one phone and directly testing on images from another produced inaccurate results. Through detailed experimentation and analysis, we found out that the above two problems impact the spectral reconstruction and classification models in different ways. They do not affect the operation of the intrinsic decomposition model. Specifically, the sensor sensitivity $\mathbf{S}$ directly impacts the training of the truthful reconstruction model, as shown by Equation (3) and Equation (4). This means

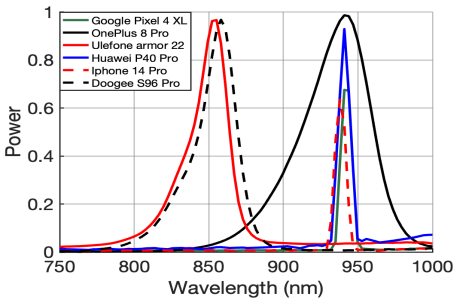

Fig. 7. Diversity of NIR systems on phones.

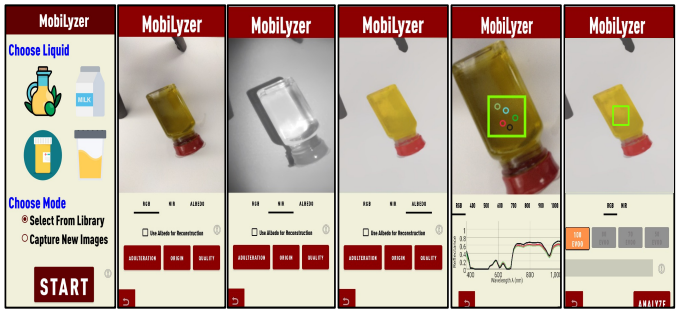

Fig. 8. Screenshots of the MobiLyzer application on Android.

that we need to train a separate reconstruction model for each phone type and model, which is not practical. To address this problem, we opted to train the reconstruction model on a wide variety of sensitivity functions and deploy the trained model on all phones. However, phone manufacturers rarely publish the sensitivity functions of their phone sensors. We found the study of [73], which experimentally measured the sensitivity function of 20 diverse phones from all major manufacturers, including Apple, Google, Samsung, Huawei, Sony, Xiaomi, Sharp, and Fujitsu. We computed the average sensitivity across all phones and used it in training our reconstruction model. Our results in §5 demonstrate that this practical approach yields accurate spectral reconstruction results across phones without requiring any training or fine-tuning.

For the second problem, we noticed that the heterogeneity of ISP operations results in shifts in the color distributions of the produced RGB images, as demonstrated in Figure 6.c. These color shifts do not impact the spectral reconstruction model, but they affect the classification model. To evaluate the robustness of our system, we tested MobiLyzer on four heterogeneous smartphones, without any phone-specific adaptation. As shown in our experiments in §5.4, MobiLyzer consistently achieves high classification accuracy across all devices, demonstrating strong generalization to various sensor and ISP configurations.

NIR Camera Systems. As described in §3.1, fine-grained spectral analysis of liquids requires information beyond the visible range that is captured by RGB cameras. Specifically, NIR information is crucial for identifying subtle molecular properties relevant to constituent analysis. However, phone manufacturers design diverse NIR systems in their phones, which, if not properly considered, would compromise the accuracy and generality of spectral analysis across various phones.

We begin by dividing the most common designs for NIR systems on phones into three categories: face recognition, night vision, and full-spectrum. In the first category, NIR-based face recognition systems have emerged as a cornerstone of modern smartphone security [72]. Most major phone manufacturers, including Apple, Samsung, Google, Huawei, and OnePlus, offer them. For example, Apple's Face ID employs a structured light system that projects approximately 30,000 IR dots at a wavelength of 940 nm to generate a detailed 3D facial map. Google's uDepth offers an active stereo IR sensor that operates at 940 nm [34].

Some smartphones feature dedicated night vision camera modules, forming the second category of mobile NIR systems. Such modules utilize sensors, such as Sony's IMX [5], which are specifically designed to capture NIR wavelengths around 850 nm [51]. The third category is devices that offer full-spectrum camera modules, such as the OnePlus 8 Pro. These camera modules do not have IR filters. Thus, the RGB bands captured by such modules are mixed with IR signals. The IR signals appear more with the R band because the wavelength of red light is closer to the NIR range. To model and mitigate the effect of heterogeneous NIR systems, we experimentally analyze representative samples from three main categories. Specifically, we analyze the following NIR systems:

- iPhone 14 Pro, Huawei P40 Pro, and Google Pixel 4XL from the face identification category,

- Doogee S96 Pro and Ulefone Armor 22 from the night vision category, and
- OnePlus 8 Pro from the full-spectrum category.

We use the hyperspectral camera mentioned in §3.1 in this analysis. For each phone, we capture the reflectance profiles of the NIR system by the hyperspectral camera. Specifically, we use a while surface that reflects all wavelengths. We operate the NIR camera of the phone and direct it towards the white surface. We capture the reflected signals using the hyperspectral camera. We plot the results in Figure 7. For the full-spectrum case (OnePlus 8 Pro), we plot the R band only, as the G and B bands have minimal NIR signals mixed with them. The results in the figure reveal two important insights. First, most NIR systems operate around two main wavelengths: 850 and 940 nm. Second, there are variations across phones that operate around each of these wavelengths.

Based on our analysis and experiments above, we address the heterogeneity of the NIR systems by training two separate reconstruction models. One for all phones with NIR systems that operate at 850 nm, and the other for phones that operate at 940 nm. Further, during training, we randomly add a Gaussian noise with a mean of 10 nm to account for the variations across different phone models.

Summary. MobiLyzer handles the diversity of RGB camera systems by training the reconstruction model on the average spectral sensitivity function derived from a wide range of phones. To address the diversity of NIR camera systems, MobiLyzer trains two separate reconstruction models: one centered on the NIR channel at 850 nm and the other at 940 nm, ensuring compatibility with the dominant NIR hardware configurations in modern phones. We note that training the reconstruction model is complex as it requires ground truth hyperspectral images. In MobiLyzer, we train only twice using the same hyperspectral dataset that we collected, and we will make it open source. As shown in our experiments in §5.4, MobiLyzer consistently achieves high classification accuracy across all devices, demonstrating strong generalization to various sensor and ISP configurations.

4.5 Implementation of the MobiLyzer App

We developed a proof-of-concept application of MobiLyzer on the Android platform; some screenshots are shown in Figure 8. The application implements all components shown in Figure 1, including the intrinsic decomposition described in §4.2, the truthful spectral reconstruction presented in §4.3, and multiple classifiers customized for different analysis tasks. We design a base one-dimensional convolutional neural network (1DCNN) classification model, which is suitable to process hyperspectral signatures represented as a 68-dimensional vector. This architecture is well-suited for learning and exploiting the sequential pattern of spectral data, as it effectively captures local correlations between adjacent wavelengths in signatures. It has also been proven to be powerful and memory-efficient in prior works [75]. We then customize this model through training to different tasks.

We present the details of optimizing the intrinsic decomposition model for phones in §A.4, and we describe our training of the reconstruction and classification models to improve their robustness and avoid overfitting in §A.2. We note that all machine learning models are first trained on a workstation. Then, the trained models are deployed to phones for inference.

As shown in Figure 8, MobiLyzer currently supports analyzing five different types of liquids: olive oil, milk, liquid medicine, urine, and honey. It can be easily extended to other liquids by fine tuning our reconstruction and classification models. The MobiLyzer application works by capturing RGB and NIR images of the liquid using phone cameras under regular illumination. It computes the albedo of the RGB image using the intrinsic decomposition model. It then calls the reconstruction model to upsample the RGB albedo and NIR to N bands. Through experimentation, we found that $N = 68$ provides sufficient accuracy for constructing spectral signatures of liquids; higher N values required more memory and computing resources for training and inference, but they did not improve the accuracy. The application has customized screens for each liquid, depending on the desired analysis tasks, e.g., quality grading, origin determination, and fraud analysis of olive oil. The application presents a simple user interface as well as detailed views where expert users can inspect and compare spectral signatures.

Table 2. Performance comparison between the proposed truthful spectral reconstruction model and the state-of-the-art.

	SAM	SID	PSNR	ΔE	MAE_{RGB}	MAE_{NIR}	Memory (MB)	Inference Time (ms)
MobiSpectral [67]	0.06	0.01	31.94	2	0.14	0.17	90	2500
TSR (this paper)	0.07	0.02	31.28	0.00	0.00	0.00	7.4	760

4.6 Limitations and Practical Considerations

MobiLyzer relies on analyzing the spectrum reflected from liquids in bottles. While many bottles are transparent, some are dark and/or covered by large paper labels, which limit light transmission. Similarly, metal containers will block the light from passing to the liquid. In such cases, small samples of the liquid will need to be poured into a transparent container for analysis. In addition, as shown in §5, MobiLyzer functions in practical environments that have diverse illuminations. However, MobiLyzer requires reasonable illumination such that there is enough reflected light from the liquid. If the environment has low lighting, the flash of the phone may be used to provide enough illumination.

Further, MobiLyzer utilizes NIR signals. While most current phones have NIR cameras, some manufacturers, e.g., Apple, do not currently allow external developers to access them. Finally, the models of MobiLyzer need to be trained using ground-truth hyperspectral images. We make our trained models, datasets, and code open-source, providing a starting point. For analyzing liquids other than the ones in our datasets, our trained models can be fine-tuned by using only a few hyperspectral images.

5 Evaluation

We first evaluate the proposed truthful spectral reconstruction method in §5.1. Then, we demonstrate the accuracy, fine-granularity, and generality of MobiLyzer in analyzing various liquids in §5.2. Then, we conduct an ablation study to analyze the performance contributions of various components in §5.3. We further demonstrate the robustness of MobiLyzer to practical illuminations, phone diversity, and capturing distances in §5.4. In the same section, we also analyze the small computational requirements, which make MobiLyzer easily deployable on current phones. **The code and datasets of this paper are open source** [50].

5.1 Performance of the Truthful Spectral Reconstruction Model

Model Compared Against. We assess the accuracy of the proposed reconstruction model (TSR) and compare it against the state-of-the-art, MobiSpectral [67]. To ensure fairness, we train both models using their fruit datasets, and we reconstruct the same number of bands (68). The fruit data set has 346 hyperspectral images of different fruits, including apples, kiwis, tomatoes, strawberries, and blueberries. We partition this dataset into 70% for training, 15% for validation, and 15% for testing.

Performance Metrics. We compare the accuracy of TSR and MobiSpectral using the same performance metrics defined in [67], including SAM, SID, and PSNR. SAM and SID measure the differences (or errors) between the reconstructed and ground-truth bands. Thus, lower SAM and SID values are better, with zero as the minimum for both metrics. PSNR measures the quality (in dB) of the reconstructed bands relative to the ground truth ones; higher values are better, and PSNR > 30 dB usually indicates good reconstruction quality. SAM, SID, and PSNR are computed across all 68 bands.

In addition, we assess the truthfulness of the reconstructed four R, G, B, and NIR bands using two additional metrics: Mean Absolute Error (MAE) and ΔE. We compute the MAE between the original and reconstructed RGB images (MAE_{RGB}), as well as between the original and reconstructed NIR images (MAE_{NIR}). Unlike SAM, SID, and PSNR, MAE focuses on the errors in the RGB and NIR bands. Further, we compute the standard color

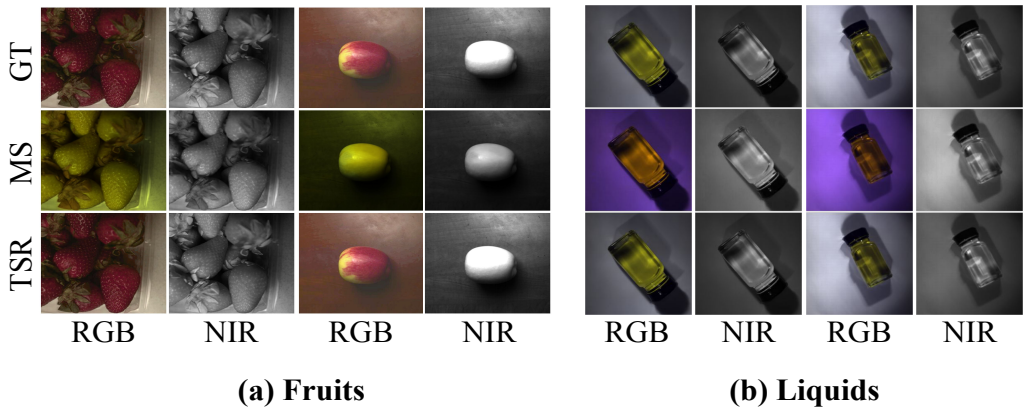

Fig. 9. Sample results demonstrating the importance of our truthful reconstruction (TSR) versus state-of-the-art (MS).

error metric ΔE, which is commonly used in industries (like printing and displays) to objectively measure color accuracy and consistency [45].

Finally, we measure the memory requirements and average inference time for both TSR and MobiSpectral when running these models on a workstation with a 12th Gen Intel Core i7-12700 processor (up to 4.9 GHz turbo), an NVIDIA GeForce RTX 3080 GPU (10 GB VRAM), and 32 GB of RAM.

Quantitative Results. We summarize the average results of all metrics in Table 2. The results show that TSR achieves truthful reconstruction: the MAE_{RGB}, MAE_{NIR}, and ΔE are all zero. In contrast, MobiSpectral introduces significant errors in reconstructing these four critical bands. For example, MobiSpectral results in $\Delta E = 2$, which is a noticeable color error [45]. This color error could compromise the accuracy of the spectral analysis of liquids. The results also show that TSR is more efficient than MobiSpectral: it requires 90% less memory and produces an inference within an average of 760 ms, compared to 2500 ms for MobiSpectral. Computational efficiency is crucial for mobile applications that run on devices with limited resources.

Visual Samples. We present sample visual results in Figure 9 to demonstrate the accuracy of the reconstructed bands by our truthful reconstruction model compared to those produced by MobiSpectral. For reference, we also include the ground-truth bands captured by a hyperspectral camera in the figure. The shown samples are from the fruits dataset in [67] and from our liquids datasets.

5.2 Performance Analysis of MobiLyzer

In this section, we demonstrate the accuracy of MobiLyzer in providing fine-grained analysis of different liquids. We consider several variations and concentrations of five representative liquids: olive oil, milk, honey, urine, and liquid medicine (Tylenol). We first present a detailed case study of analyzing olive oil in terms of detecting adulteration, assessing quality, and determining the country of origin. Then, we analyze different types of dairy and non-dairy milks with various fat percentages. Finally, we demonstrate the generality of MobiLyzer by analyzing the three other liquids.

Experimental Setup. Our testbed includes the following:

- ***Four Diverse Phones.*** We use four unmodified mobile phones: Google Pixel 4XL, Doogee S96 Pro, Ulefone Armor 22, and OnePlus 8 Pro. All phones have NIR cameras for different purposes in addition to the regular RGB cameras. For example, Google Pixel 4XL uses NIR at 940 nm for face identification. Doogee and Ulefone

Fig. 10. Samples of our mobile image dataset captured using diverse phones and illuminations for various liquids and containers.

utilize NIR at 850 nm for night vision, whereas OnePlus features a full-spectrum camera that combines NIR signals with RGB bands. These four phones cover the three categories of the common NIR camera systems described in §4.4. They all have different sensor sensitivity functions and image processing pipelines.

- ***High-end Hyperspectral Camera.*** We use the Specim IQ hyperspectral camera mentioned in §3.1. It captures 204 bands in the VNIR range, each band has a resolution of 512 x 512 pixels.
- ***Seven Illumination Sources.*** To account for the diversity of illuminations in real environments, we experiment with *seven* illumination sources: Halogen, Fluorescent, LED Cool, LED Warm, LED Neutral, Multiple (comprising all the previous ones), and Arbitrary. In the Arbitrary setting, we do not control the illumination; instead, we capture using our normal lab lighting and sunlight coming from the windows.
- ***Five Liquids with Various Mixtures and Concentrations.*** We analyze liquids with heterogeneous properties, e.g., viscosity and color, and consider different scenarios, including: (i) olive oil (multiple qualities and origins), (ii) milk (from lactose free to full fat), (iii) honey (pure and mixed with corn syrup), (iv) liquid medicine (authentic and adulterated Tylenol), and (v) urine (various concentrations of protein).
- ***Diverse Container Shapes and Materials.*** To further stress MobiLyzer and assess its performance in realistic settings, we purchased different types and sizes of containers, including glass and plastic ones.

We captured two datasets:

- ***Hyperspectral Images Dataset.*** This dataset is captured using the hyperspectral camera under halogen illumination. It has a total of 770 hyperspectral images: (i) 380 for olive oil, (ii) 200 for milk, (iii) 90 for urine, (iv) 60 for liquid medicine, and (v) 40 for honey. Each hyperspectral image contains 204 spectral bands with a spatial resolution of 512 x 512 pixels. This dataset is divided into two partitions: (i) 70% used to train the spectral reconstruction model and (ii) 30% used to evaluate its performance, 15% for evaluation, and 15% for testing.
- ***Mobile Images Dataset.*** This dataset is captured using the four phones and under seven diverse illuminations mentioned above. It has RGB and NIR image pairs. The total number of image pairs in this dataset is 2,150, with the following distribution: (i) 1300 for olive oil, (ii) 350 for milk, (iii) 210 for liquid medicine, (iv) 140 for honey, and (v) 150 for urine. For each liquid category, images were captured under seven diverse illumination conditions, with each illumination setting including 10 unique images. Importantly, the illumination angle and the type of illumination vary for each individual image, ensuring substantial diversity in lighting conditions across the dataset. Samples of this dataset are shown in Figure 10. The figure shows the diversity of images in terms of liquids, illuminations, containers, backgrounds, and capturing distances and angles. This dataset is used to assess the end-to-end accuracy of MobiLyzer.

Training of Machine Learning Models. MobiLyzer has the following three models:

- ***Intrinsic Decomposition.*** As detailed in §A.4, we optimize the pre-trained model in [13] for mobile platforms. This model is used for all phones and liquids, without any further configuration or fine-tuning.
- ***Spectral Reconstruction.*** We train a spectral reconstruction model for each liquid to recover the full VNIR spectrum. During training, the model takes as input synthetic RGB-NIR images derived from hyperspectral measurements, and uses the original hyperspectral data as ground truth. To ensure color consistency, we enforce color fidelity constraints that preserve the formation of the input channels. To support deployment across different smartphone hardware, we train two versions of the model: one for all phones with NIR signals at 850 nm and the other for phones with NIR signals at 940 nm.
- ***Classification.*** We train a separate version of this model for each liquid analysis task considered, e.g., identifying the country of origin of olive oil. The mobile images dataset is used in this training.

We describe the details of training the reconstruction and classification models in §A.2.

Fine-Grained Fraud Analysis of Olive Oil. Olive oil is reported to have multiple health benefits [48], but it comes at a significantly higher price than other oils. This makes olive oil one of the most frequently adulterated products. Adulteration typically occurs with cheaper oils that look somewhat similar, such as peanut oil. Detecting such adulteration is essential to protect consumers and honest producers. Also, adulteration may pose health hazards. For example, some consumers may be allergic to peanuts, and mixing peanut oil with olive oil could pose a significant health risk.

We purchased authentic extra virgin olive oil (referred to as 100% EVOO) from a trusted local store. We then mixed samples from this olive oil with peanut oil, in controlled ratios by volume: (i) 50% peanut oil added to EVOO (50% EVOO), (ii) 30% peanut oil added to EVOO (70% EVOO), and (iii) 20% peanut oil added to EVOO (80% EVOO). We use MobiLyzer to detect this adulteration with different ratios using the olive oil images of the mobile images dataset described above. There are 560 images of olive oil adulteration scenarios: (i) 140 for 100% EVOO, 140 for 80% EVOO, 140 for 70% EVOO, and 140 for 50% EVOO.

We begin by partitioning the dataset into two subsets: 75% for training and 25% as a fixed hold-out test set. This test set remains entirely unseen during training the classifier and is reserved exclusively for final model evaluation. On the training subset, we perform 4-fold cross-validation to ensure that class distributions are preserved across all folds. We compute and report standard classification metrics, including accuracy, precision, and recall. We report the results in Figure 11, which shows the high accuracy of detecting even small percentages (20% and 30%) of adulteration. The average accuracy across all cases is 96%.

Identifying Grade and Origin of Olive Oil. Because of the climate conditions, soil composition, and long-established processes, olive oil from specific countries, e.g., Italy, is more expensive than from other countries. Thus, a common fraud case involves incorrectly changing the country of origin. In addition, olive oil comes in different grades: EVOO, Refined (ROO), and Pomace (POO). EVOO is the most expensive, whereas POO is the least. Mislabeling a lower grade as a more expensive one is also a common cheating mechanism.

We purchased samples of olive oil from Italy and California (USA). We also purchased samples from the three grades of olive oil (EVOO, ROO, and POO). Then, we used MobiLyzer to identify the origin country and grade of olive oil. We note that while various grades of olive oil may look similar, they have different concentrations of fatty acids [32]. The country of origin also has a similar effect on the chemical composition of olive oil [41]. Temperature and rainfall are the key geographical factors that control olive oil's chemical composition, particularly fatty acid profiles and bioactive compounds. Geographic origin was identified as the main contributor to variation in oleic and linoleic fatty acids, with higher temperatures and drier conditions significantly influencing the oleic/linoleic ratio [60]. Lower temperatures and higher rainfall have also been shown to slow down fruit growth and development, ultimately producing olive oils with different fatty acid compositions and antioxidant components [46].

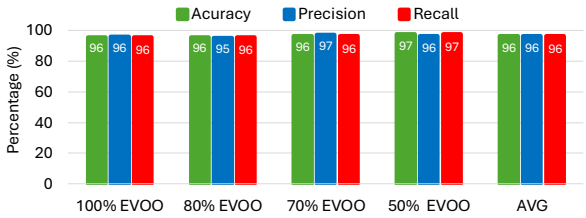

Fig. 11. Detecting adulteration in olive oil.

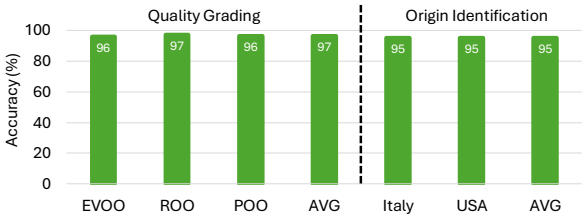

Fig. 12. Identifying grades and origin of olive oil.

Table 3. Comparison between MobiLyzer and recent works for analyzing olive oil.

Work	Experiment	Accuracy
RF-EATS [74]	Identify 20% EVOO	~75%
LiquidHash [68]	Identify 80% EVOO	~64%
	Identify 70% EVOO	~80%
	Identify 50% EVOO	~82%
LiDAR [18]	Distinguish OO from other liquids	~79%
[37, 59, 65, 69, 80]	Did not (or cannot) analyze OO	—
MobiLyzer	Detect adulteration, origin, grade	>95%

We summarize the results in Figure 12. Although the visual and chemical differences between olive oil from Italy and USA are very subtle, MobiLyzer can correctly identify the country of origin with an average accuracy of 95%. Similarly, MobiLyzer can differentiate among the grades of olive oil with an average accuracy of 97%. This is enabled by the detailed spectral analysis it performs, which utilizes the RGB albedo and NIR images and upsamples them by a truthful reconstruction model.

Comparing MobiLyzer vs. Others for Analyzing Olive Oil. We compare the performance of MobiLyzer against the closest works in the literature that analyzed olive oil. Some of these works, such as RF-EATS [74], require specialized hardware. Others, like LiDAR [18], focus on simpler cases, such as distinguishing olive oil from completely different liquids. Most of these works did not release their codes and datasets, making it hard to reproduce their results. Thus, we conduct *approximate* comparisons using the *best results* reported in their papers.

Table 3 summarizes the experiments and results reported in [18, 68, 74] and compares them against what MobiLyzer achieves. We note that x% EVOO means mixing x% of extra virgin oil with $(100 - x)$% peanut oil. The table indicates that MobiLyzer is more general as it can identify the origin and grade of olive oil and detect various degrees of adulteration. In addition, the accuracy of MobiLyzer is much higher than previous works, including the one that requires extra hardware [74]. The authors of RF-EATS [74] note that the reason for its lower accuracy in classifying olive oil adulteration is the similar permittivity properties of peanut and olive oils. Similarly, the accuracy of LiquidHash [68] drops significantly with small adulteration percentages because olive and peanut oils have close viscosity coefficients. In contrast, MobiLyzer analyzes the chemical compositions of liquids using spectral analysis, which enables accurate and finer-grained analysis.

Fine-Grained Analysis of Milk. We assess the performance of MobiLyzer for fine-grained analysis of different types of milk. Some previous studies, like [59], involve the use of sophisticated hardware. Other works, such as [69], only compare skim and whole milk, which have the largest differences in fat content and thus much easier to differentiate. To demonstrate the fine granularity of our approach, we show that MobiLyzer can distinguish

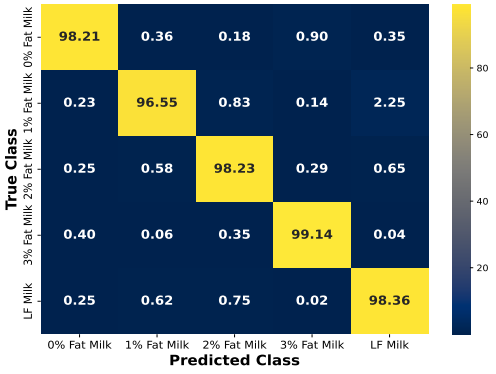

Fig. 13. Accuracy of identifying milk types.

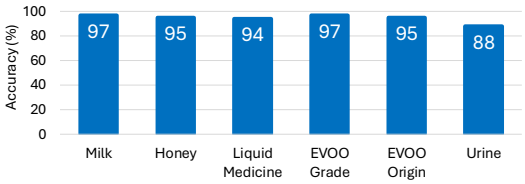

Fig. 14. Accuracy of identifying various liquids.

milk samples with as little as 1% differences in fat content. To further highlight the granularity and versatility of MobiLyzer, we include 1% fat lactose-free (LF), which is crucial for individuals with dietary restrictions.

We summarize the analysis results in Figure 13, which show the confusion matrix produced by our system when analyzing five classes of milk: 0%, 1%, 2%, 3%, and LF. As shown in the figure, MobiLyzer achieves high accuracy in all cases, with an average accuracy of 97%, demonstrating its granularity and effectiveness.

Analyzing Other Liquids: Honey, Urine, and Liquid Medicine. We demonstrate the generality of MobiLyzer by analyzing three other liquids with different characteristics and addressing important problems: Honey, Urine, and Liquid Medicine. For honey, we distinguish between 100% pure honey and honey adulterated with 20% cheap corn syrup. As Figure 14 shows, MobiLyzer achieves a high accuracy of 95%.

We synthesized artificial urine by mixing distilled water with urea (commonly used as an agricultural nitrogen fertilizer), following the approach in [37]. The urea concentration was set to approximately 38 mg/100 mL, which is typical for healthy individuals. To simulate different protein levels, we introduce three concentrations of protein powder into the artificial urine: 10 mg/100 mL to represent healthy individuals, 15 mg/100 mL to model a person at the border of potentially having kidney disease, and 30 mg/100 mL to represent a person with a higher chance of kidney disease. The latter corresponds to the condition of microalbuminuria [78]. We then assess the MobiLyzer's accuracy in distinguishing between healthy urine samples and those indicating potential microalbuminuria. As Figure 14 shows, MobiLyzer achieves a high accuracy of 88% even in this challenging case.

Finally, we analyzed liquid medicine. Counterfeit medicine is a serious problem in many countries. In a recent case, fake cough syrup contained 90% diethylene glycol, a toxic chemical found in antifreeze [16]. To create a similar counterfeit sample, we removed half of the contents from a Tylenol bottle and replaced it with diethylene glycol for testing. MobiLyzer can accurately detect this cheating case as well, as shown in Figure 14.

Summary. The results in this section demonstrate the generality, fine-granularity, and accuracy of MobiLyzer in analyzing different liquids and addressing various problems.

5.3 Ablation Study

We analyze the performance impact of different components of MobiLyzer. Specifically, we evaluate the accuracy when using only RGB images for liquid analysis. Next, we add NIR images to the RGB image and assess the accuracy. Finally, we use all components of MobiLyzer, including the spectral reconstruction model, which usamples the RGB and NIR images to multiple spectral bands.

Figure 15 summarizes the results of our ablation study for two sample liquids: olive oil and milk. As the figure shows, RGB images alone provide low accuracy because they can only detect color differences in liquids. Most

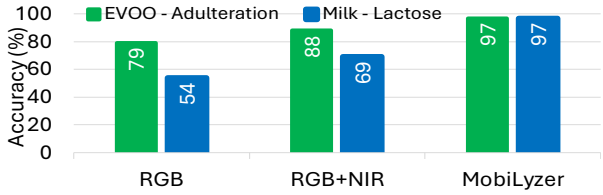

Fig. 15. Ablation study: Impact of different components.

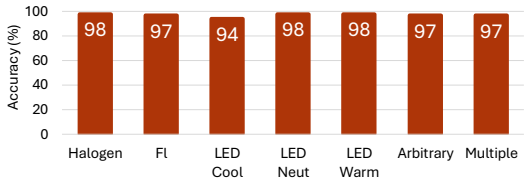

Fig. 16. Robustness to practical illuminations.

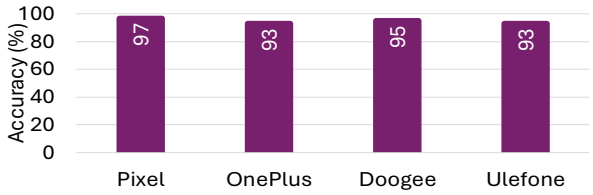

Fig. 17. Robustness to phone diversity.

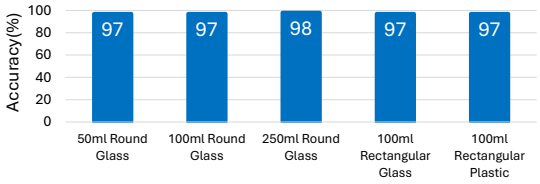

Fig. 18. Robustness to container shape and material.

fraud cases or subtle chemical composition differences are not easily detectable in the visible range. Using NIR images improves performance, as NIR signals can penetrate liquid surfaces and reveal some information about their chemical compositions. However, NIR images captured by phones cover only one narrow spectral band, providing limited information to the classification model. The truthful spectral reconstruction model of MobiLyzer uses the RGB bands along with the NIR one and creates many spectral bands across the entire 400–1000 nm range. This reveals much more information about the chemical composition of different liquids and allows for computing more accurate spectral signatures to represent them in a high-dimensional space. These feature-rich spectral signatures significantly improve the accuracy as shown in Figure 15.

5.4 Robustness and Computational Complexity

Robustness to Illumination Diversity. Our mobile images dataset was captured under diverse lighting conditions, with illumination settings ranging from low to high color temperatures to cover a wide range of scenarios. Additionally, we included arbitrary and multiple illumination settings to represent complex practical environments. We assess the accuracy under different illuminations by separating the images for each illumination. We present the results in Figure 16, which demonstrate the accuracy of MobiLyzer under diverse illuminations. MobiLyzer achieves this robustness because of its intrinsic decomposition model that mitigates the effect of illumination and the truthful reconstruction model that preserves the original colors in the reconstructed bands.
Robustness to Phone Diversity. In §4.4, we analyzed various aspects of phone diversity, including sensor sensitivity, processing pipeline, and NIR design. We assess the accuracy of MobiLyzer across four heterogeneous phones in all of these aspects. The results, shown in Figure 17, confirm the robustness and versatility of MobiLyzer, as we observe consistent and high accuracy across all phones, confirming the robustness and generalization capability of MobiLyzer.
Robustness to Container Shape and Material. We investigate the robustness of MobiLyzer against variations in container size, shape, and material. Specifically, we evaluate five different containers: 50 ml round glass, 100 ml round glass, 250 ml round glass, 100 ml rectangular plastic, and 100 ml rectangular glass. Our results in Figure 18 show consistently high accuracy across all tested containers, with accuracy values ranging from 97% to 98%. This robustness is achieved due to the intrinsic decomposition component, which mitigates these effects.
Robustness to Capturing Distance. We investigate how varying the capturing distance between the smartphone and the liquid sample affects the accuracy of MobiLyzer. Specifically, we evaluate distances ranging from 10 cm

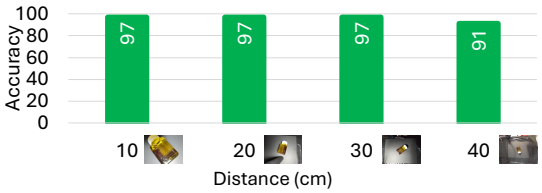

Fig. 19. Robustness to capturing distances.

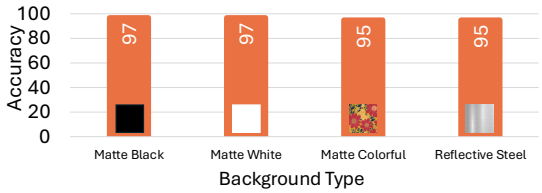

Fig. 20. Robustness to different backgrounds.

to 40 cm, which cover the practical operational range of typical NIR smartphone cameras. Our results, depicted in Figure 19, show that MobiLyzer consistently achieves high accuracy (97%) for capturing distances between 10 cm and 30 cm. However, the accuracy started to decrease at 40 cm. This decline is attributed to the weaker NIR signal intensity at distances $\geq$ 40 cm, leading to less precise spectral reconstruction. Overall, MobiLyzer remains robust within the practical capturing range of NIR cameras, providing flexibility in real-world scenarios.

We note that the range of RGB cameras are much larger (order of meters) than that of NIR cameras. That is, the NIR cameras are the limiting factor for MobiLyzer. In addition, the viewing angle does not impact the accuracy of MobiLyzer. This is indicated by the results in Figure 19, where we show the corresponding RGB image for each capturing distance. In our datasets, we captured images if liquid containers in different/arbitrary directions and angles. The reason behind the limited impact of the viewing angle is that we conduct the spectral analysis on a small rectangular area of the captured scene. The MobiLyzer application automatically chooses this area in the middle of the image, but it allows the user to move it to focus mostly on the liquid.

Robustness to Background. We analyze the effect of different background types on the performance of MobiLyzer. Specifically, we tested four distinct backgrounds: matte black, matte white, matte colorful, and reflective steel. Our results in Figure 20 indicate high accuracy (97%) with neutral matte backgrounds (black and white). However, the accuracy slightly decreases to 95% for the matte colorful and reflective steel backgrounds, due to the increased variability in reflections and lighting conditions affecting the captured signals. Nonetheless, these accuracy values remain high, confirming that MobiLyzer effectively minimizes background interference, enabling it to reliably function across various practical scenarios.

Computational Complexity. We measure the memory requirements and running time of MobiLyzer on two different phones: Google Pixel 4 XL (released in 2019) and OnePlus 8 Pro (released in 2020). Both are relatively old with limited resources compared to recent phones. We summarize our results in Table 6 for the Pixel phone; §A.5 contains the results for the other phone. For the quantized versions of our models, the table shows that the end-to-end running time of MobiLyzer is 6.15 seconds, on average. This is the total time between capturing the RGB and NIR images and producing the final results. On the OnePlus 8 Pro phone, which has slightly more computing power, our measurements indicate that the running time is 5.57 seconds. On recent phones, the running time is expected to be even lower. The total memory required to deploy MobiLyzer is 329 MB on both phones.

6 Conclusion

We presented a mobile system, MobiLyzer, that utilizes the sensing capabilities of recent phones and machine learning models to analyze various aspects of liquids. It enables accurate spectral analysis of liquids on unmodified phones and is robust to the diversity of camera hardware and illumination. This spectral analysis reveals subtle differences among liquids based on how they reflect different wavelengths. We presented methods to mitigate the effect of illumination and interference from liquid containers. We extended prior spectral reconstruction models to improve their accuracy, robustness, and efficiency, making them practical for liquid analysis and

Table 4. Inference time and memory requirements of MobiLyzer on Google Pixel 4XL.

	No Quantization		With Quantization	
	Time (s)	Memory (MB)	Time (s)	Memory (MB)
Intrinsic Decomposition (Stage 1)	3.04	462.25	1.86	118.11
Intrinsic Decomposition (Stage 2)	2.37	398.39	1.17	101.10
Intrinsic Decomposition (Stage 3)	4.91	398.40	2.35	101.08
Intrinsic Decomposition (Total)	10.32	1259.04	5.38	320.29
Spectral Reconstruction	0.76	7.4	0.76	7.4
Classification	0.01	1.3	0.01	1.3
Total (End-to-End)	**11.09**	**1267.74**	**6.15**	**328.99**

other future mobile applications that may benefit from spectral analysis. We also optimized the state-of-the-art intrinsic decomposition model to make it deployable on the resource-constrained mobile platforms, which could potentially enable many other mobile applications beyond liquid analysis.

We implemented MobiLyzer on the Android platform and conducted detailed experimentation with multiple liquids, including olive oil, milk, liquid medicine, honey, and urine. We considered important problems, such as early detection of kidney diseases and detection of adulteration in liquid medicine. We demonstrated the ability to perform fine-grained analysis of liquids, e.g., identifying quality grades and country of origin of olive oil. Our results confirmed the accuracy, robustness, and versatility of MobiLyzer. For example, MobiLyzer can detect adulteration in olive oil with an average accuracy of 96% even for small (e.g., 20%) adulteration ratios. It can also identify the grade and country of origin of olive oil with an average accuracy of 97% and 95%, respectively. Furthermore, these results were obtained on regular phones operating in realistic environments with diverse illuminations.

References

[1] Nour Aburaed, Mohammed Q. Alkhatib, Stephen Marshall, Jaime Zabalza, and Hussain Al Ahmad. 2022. A Comparative Study of Loss Functions for Hyperspectral SISR. In *2022 30th European Signal Processing Conference (EUSIPCO)*. 484–487. doi:10.23919/EUSIPCO55093.2022.9909827

[2] Mahmoud Afifi, Jonathan T Barron, Chloe LeGendre, Yun-Ta Tsai, and Francois Bleibel. 2021. Cross-camera convolutional color constancy. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 1981–1990.

[3] Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. 2019. Optuna: A Next-generation Hyperparameter Optimization Framework. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* (Anchorage, AK, USA) *(KDD '19)*. Association for Computing Machinery, New York, NY, USA, 2623–2631. doi:10.1145/3292500.3330701

[4] Jarmo T Alander, Vladimir Bochko, Birgitta Martinkauppi, Sirinnapa Saranwong, and Timo Mantere. 2013. A Review of Optical Nondestructive Visual and Near-Infrared Methods for Food Quality and Safety. *International Journal of Spectroscopy* 2013, 1 (Jan. 2013), 341402.

[5] M. R. Alarcón, J. Licandro, M. Serra-Ricart, E. Joven, V. Gaitan, and R. d. Sousa. 2023. Scientific cmos sensors in astronomy: imx455 and imx411. *Publications of the Astronomical Society of the Pacific* 135 (2023), 055001. Issue 1047. doi:10.1088/1538-3873/acd04a

[6] Boaz Arad and et al. 2022. NTIRE 2022 Spectral Recovery Challenge and Data Set. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR'22) Workshops*. 863–881.

[7] Sean Bell, Kavita Bala, and Noah Snavely. 2014. Intrinsic images in the wild. *ACM Transactions on Graphics (TOG)* 33, 4 (2014), 1–12.

[8] E Berra, S Gibson-Poole, A MacArthur, R Gaulton, and A Hamilton. 2015. ESTIMATION OF THE SPECTRAL SENSITIVITY FUNCTIONS OF UN-MODIFIED AND MODIFIED COMMERCIAL OFF-THE-SHELF DIGITAL CAMERAS TO ENABLE THEIR USE AS A MULTISPECTRAL IMAGING SYSTEM FOR UAVS. *Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci.* XL-1/W4 (Aug. 2015), 207–214.

[9] Nicolas Bonneel, Balazs Kovacs, Sylvain Paris, and Kavita Bala. 2017. Intrinsic Decompositions for Image Editing. *Computer Graphics Forum (Eurographics State of The Art Report)* (2017).

[10] Yuanhao Cai, Jing Lin, Zudi Lin, Haoqian Wang, Yulun Zhang, Hanspeter Pfister, Radu Timofte, and Luc Van Gool. 2022. MST++: Multi-Stage Spectral-Wise Transformer for Efficient Spectral Reconstruction. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR'22) Workshops*. 745–755.

[11] Canadian Food Inspection Agency. 2022. Food Fraud Annual Report 2021 to 2022. https://inspection.canada.ca/en/science-and-research/our-research-and-publications/food-fraud-report

[12] Chris Careaga and Yağız Aksoy. 2023. Intrinsic Image Decomposition via Ordinal Shading. *ACM Trans. Graph.* 43, 1, Article 12 (2023), 24 pages.

[13] Chris Careaga and Yağız Aksoy. 2024. Colorful Diffuse Intrinsic Image Decomposition in the Wild. *ACM Trans. Graph.* 43, 6, Article 178 (2024), 12 pages.

[14] Chris Careaga and Yağız Aksoy. 2024. Colorful Diffuse Intrinsic Image Decomposition in the Wild. https://github.com/compphoto/Intrinsic. accessed on 2025-07-30.

[15] Michael F Carroll and Jonathan L Temte. 2000. Proteinuria in adults: a diagnostic approach. *American family physician* 62, 6 (2000), 1333–1340.

[16] Centers for Disease Control and Prevention (CDC). 2009. Fatal poisoning among young children from diethylene glycol-contaminated acetaminophen - Nigeria, 2008-2009. *MMWR Morb. Mortal. Wkly. Rep.* 58, 48 (Dec. 2009), 1345–1347.

[17] Justin Chan, Anandithaa Raghunath, Kelly E. Michaelsen, and Shyamnath Gollakota. 2022. Testing a Drop of Liquid Using Smartphone LiDAR. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 6, 1, Article 3 (mar 2022), 27 pages. doi:10.1145/3517256

[18] Justin Chan, Anandithaa Raghunath, Kelly E. Michaelsen, and Shyamnath Gollakota. 2022. Testing a Drop of Liquid Using Smartphone LiDAR. In *Proceedings of ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies (IMWUT'22)*.

[19] Chein-I Chang. 2003. *Hyperspectral Imaging: Techniques for Spectral Detection and Classification*.

[20] Baicheng Chen, John Nolan, and Xinyu Zhang. 2024. MetaBioLiq: A Wearable Passive Metasurface Aided mmWave Sensing Platform for BioFluids. In *Proceedings of the 30th Annual International Conference on Mobile Computing and Networking* (Washington D.C., DC, USA) *(ACM MobiCom '24)*. Association for Computing Machinery, New York, NY, USA, 1192–1206. doi:10.1145/3636534.3690687

[21] J. B. Cohen and W. E. Kappauf. 1982. Metameric Color Stimuli, Fundamental Metamers, and Wyszecki's Metameric Blacks. *American Journal of Psychology* 95 (1982), 537–564.

[22] Maryam Mohammadzadeh Darrodi, Graham Finlayson, Teresa Goodman, and Michal Mackiewicz. 2015. Reference data set for camera spectral sensitivity estimation. *J. Opt. Soc. Am. A* 32, 3 (March 2015), 381–391.

[23] Partha Das, Sezer Karaoglu, and Theo Gevers. 2022. Intrinsic image decomposition using physics-based cues and CNNs. *Computer Vision and Image Understanding* 223 (2022), 103538. doi:10.1016/j.cviu.2022.103538

[24] ONNX Runtime developers. 2021. ONNX Runtime. https://onnxruntime.ai/. Version: x.y.z.

[25] M Dhanalakshmi, S Balakrishnan, A Sangeetha, K Manimaran, and V Jayalalitha. 2020. Assessment of milk adulteration in the commercially available milk for the consumers in Cauvery delta region of Tamil Nadu, India. *Indian Journal of Dairy Science* 73, 2 (May 2020). https://epubs.icar.org.in/index.php/IJDS/article/view/93856

[26] Ashutosh Dhekne, Mahanth Gowda, Yixuan Zhao, Haitham Hassanieh, and Romit Roy Choudhury. 2018. LiquID: A wireless liquid identifier. In *Proceedings of ACM Conference on Mobile Systems, Applications, and Services (MobiSys'18)*. 442–454.

[27] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. 2021. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In *Proceedings of International Conference on Learning Representations (ICLR'21)*.

[28] U.S. Food and Drug Administration. 2023. Economically Motivated Adulteration (Food Fraud). https://www.fda.gov/food/compliance-enforcement-food/economically-motivated-adulteration-food-fraud/

[29] Ying Fu, Yongrong Zheng, Lin Zhang, and Hua Huang. 2018. Spectral Reflectance Recovery From a Single RGB Image. *IEEE Transactions on Computational Imaging* 4, 3 (7 2018), 382–394. doi:10.1109/tci.2018.2855445

[30] Elena Garces, Carlos Rodriguez-Pardo, Dan Casas, and Jorge Lopez-Moreno. 2022. A Survey on Intrinsic Images: Delving Deep into Lambert and Beyond. *International Journal of Computer Vision* 130, 3 (March 2022), 836–868.

[31] E. Garcés, C. Rodríguez-Pardo, D. Casas, and J. López-Moreno. 2022. A survey on intrinsic images: delving deep into lambert and beyond. *International Journal of Computer Vision* 130 (2022), 836–868. Issue 3. doi:10.1007/s11263-021-01563-8

[32] Said Gharby, Ahmed Hajib, Mohammed Ibourki, El Hassan Sakar, Issmail Nounah, Hamza EL Moudden, Mohamed Elibrahimi, and Hicham Harhar. 2021. Induced changes in olive oil subjected to various chemical refining steps: A comparative study of quality indices, fatty acids, bioactive minor components, and oxidation stability kinetic parameters. *Chemical Data Collections* 33 (2021), 100702. doi:10.1016/j.cdc.2021.100702

[33] Silvia Ghisoni, Luigi Lucini, Federica Angilletta, Gabriele Rocchetti, Daniela Farinelli, Sergio Tombesi, and Marco Trevisan. 2019. Discrimination of extra-virgin-olive oils from different cultivars and geographical origins by untargeted metabolomics. *Food Research International* 121 (2019), 746–753. doi:10.1016/j.foodres.2018.12.052

[34] Google Research. 2020. uDepth: Real-time 3D Depth Sensing on the Pixel 4. https://research.google/blog/udepth-real-time-3d-depth-sensing-on-the-pixel-4/. Accessed: 2025-04-15.

[35] Fataneh Hashempour-baltork, Somaye Vali Zade, Yeganeh Mazaheri, Adel Mirza Alizadeh, Hossein Rastegar, Ziba Abdian, Mohammadali Torbati, and Sodeif Azadmard Damirchi. 2024. Recent methods in detection of olive oil adulteration: State-of- the-Art. *Journal of Agriculture and Food Research* 16 (2024), 101123. doi:10.1016/j.jafr.2024.101123

[36] Haiyan Hu, Yinan Zhu, Baichen Yang, Hua Kang, Shanwen Chen, and Qian Zhang. 2024. MeatSpec: Enabling Ubiquitous Meat Fraud Inspection through Consumer-Level Spectral Imaging. In *Proceedings of ACM Conference on Mobile Computing and Networking (MobiCom'24)*. 861–874.

[37] Yongzhi Huang, Kaixin Chen, Yandao Huang, Lu Wang, and Kaishun Wu. 2021. Vi-liquid: Unknown liquid identification with your smartphone vibration. In *Proceedings of ACM Conference on Mobile Computing and Networking (MobiCom'21)*. 174–187.

[38] JetBrains. 2011. Kotlin Programming Language. https://kotlinlang.org. Version accessed: July 2025.

[39] Seon Joo Kim, Hai Ting Lin, Zheng Lu, Sabine Süsstrunk, Stephen Lin, and Michael S. Brown. 2012. A New In-Camera Imaging Model for Color Computer Vision and Its Application. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 34, 12 (2012), 2289–2302. doi:10.1109/TPAMI.2012.58

[40] Gudrun J. Klinker, Steven A. Shafer, and Takeo Kanade. 1990. A Physical Approach to Color Image Understanding. *International Journal of Computer Vision* 4, 1 (1990), 7–38. doi:10.1007/BF00126537

[41] Xue Li, Yu Zhang, Zhi Liu, Wei Wang, Sulin Sun, Junhong Wang, Zuoyi Zhu, Jun Liu, Hua Yang, Shenlong Zhu, Erli Niu, and Romero Agusti. 2022. Quality assessment and geographical origin classification of extra-virgin olive oils imported into China. *Journal of Food Composition and Analysis* 113 (Oct. 2022), 104713.

[42] Zhengqi Li and Noah Snavely. 2018. Learning intrinsic image decomposition from watching the world. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 9039–9048.

[43] Yumeng Liang, Anfu Zhou, Huanhuan Zhang, Xinzhe Wen, and Huadong Ma. 2021. FG-LiquID: A Contact-less Fine-grained Liquid Identifier by Pushing the Limits of Millimeter-wave Sensing. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 5, 3, Article 116 (Sept. 2021), 27 pages. doi:10.1145/3478075

[44] Yi Tun Lin and Graham D. Finlayson. 2020. Physically plausible spectral reconstruction. *Sensors* 20, 21 (2020), 1–25. doi:10.3390/s20216399

[45] M. Ronnier Luo, G. Cui, and B. Rigg. 2001. The development of the CIE 2000 colour-difference formula: CIEDE2000. *Color Research Application* 26, 5 (2001), 340–350.

[46] Rocco Mafrica, Amalia Piscopo, Alessandra De Bruno, and Marco Poiana. 2021. Effects of Climate on Fruit Growth and Development on Olive Oil Quality in Cultivar Carolea. *Agriculture* 11, 2 (2021). doi:10.3390/agriculture11020147

[47] Zhalaliddin Makhammajanov, Abduzhappar Gaipov, Askhat Myngbay, Rostislav Bukasov, Mohamad Aljofan, and Mehmet Kanbay. 2024. Tubular toxicity of proteinuria and the progression of chronic kidney disease. *Nephrology Dialysis Transplantation* 39, 4 (April 2024), 589–599.

[48] Miguel A. Martínez-González, Carmen Sayón-Orea, Vanessa Bullón-Vela, Maira Bes-Rastrollo, Fernando Rodríguez-Artalejo, María José Yusta-Boyo, and Marta García-Solano. 2022. Effect of olive oil consumption on cardiovascular disease, cancer, type 2 diabetes, and all-cause mortality: A systematic review and meta-analysis. *Clinical Nutrition* 41, 12 (2022), 2659–2682.

[49] Maninder Meenu, Qianxi Cai, and Baojun Xu. 2019. A critical review on analytical techniques to detect adulteration of extra virgin olive oil. *Trends in Food Science Technology* 91 (2019), 391–408. doi:10.1016/j.tifs.2019.07.045

[50] Shahrzad Mirzaei, Mariam Bebawy, Amr Mohamed Sharafeldin, and Mohamed Hefeeda. 2025. *MobiLyzer Source Code, and Dataset.* https://github.com/chercode/mobilyzer-imwut25

[51] Muyao Niu, Zhuoxiao Li, Yifan Zhan, Huy H. Nguyen, Isao Echizen, and Yinqiang Zheng. 2023. Physics-Based Adversarial Attack on Near-Infrared Human Detector for Nighttime Surveillance Camera Systems. In *Proceedings of the 31st ACM International Conference on Multimedia* (Ottawa ON, Canada) *(MM '23)*. Association for Computing Machinery, New York, NY, USA, 8799–8807. doi:10.1145/3581783.3612082

[52] World Health Organization. 2017. *A Study on the Public Health and Socioeconomic Impact of Substandard and Falsified Medical Products.* https://www.who.int/publications/i/item/9789241513432/

[53] World Health Organization. 2022. Medical Product Alert N°7/2022: Substandard (contaminated) Paediatric Liquid Dosage Medicines. https://www.who.int/news/item/02-11-2022-medical-product-alert-n-7-2022-substandard-(contaminated)-paediatric-liquid-dosage-medicines/

[54] World Health Organization. 2023. WHO urges action to protect children from contaminated medicines. https://www.who.int/news/item/23-01-2023-who-urges-action-to-protect-children-from-contaminated-medicines/

[55] Brian G Osborne. 2006. Near-Infrared Spectroscopy in Food Analysis. In *Encyclopedia of Analytical Chemistry.*

[56] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. PyTorch: An imperative style, high-performance deep learning library. arXiv preprint arXiv:1912.01703. 12 pages; NeurIPS 2019.

[57] AG Pereira, P Otero, M Fraga-Corral, P Garcia-Oliveira, M Carpena, MA Prieto, and J Simal-Gandara. 2021. State-of-the-Art of Analytical Techniques to Determine Food Fraud in Olive Oils. *Foods* 10, 3 (2021), 484. doi:10.3390/foods10030484

[58] Ruiliang Pu. 2017. *Hyperspectral Remote Sensing: Fundamentals and Practices.* Routledge.

[59] Tauhidur Rahman, Alexander T. Adams, Perry Schein, Aadhar Jain, David Erickson, and Tanzeem Choudhury. 2016. Nutrilyzer: A mobile system for characterizing liquid food with photoacoustic effect. In *Proceedings of ACM Conference on Embedded Networked Sensor Systems (SenSys'16)*. 123–136.

[60] Raquel Rey-Giménez and Ana Cristina Sánchez-Gimeno. 2024. Effect of cultivar and environment on chemical composition and geographical traceability of Spanish olive oils. *Journal of the American Oil Chemists' Society* 101, 4 (2024), 371–382. arXiv:https://aocs.onlinelibrary.wiley.com/doi/pdf/10.1002/aocs.12774 doi:10.1002/aocs.12774

[61] Donghyun Ryoo, Jinyoung Hwang, and Hoeil Chung. 2017. Probing temperature able to improve Raman spectroscopic discrimination of adulterated olive oils. *Microchemical Journal* 134 (2017), 224–229. doi:10.1016/j.microc.2017.06.002

[62] Shreya N Sahasrabudhe, Veronica Rodriguez-Martinez, Meghan O'Meara, and Brian E Farkas. 2017. Density, viscosity, and surface tension of five vegetable oils at elevated temperatures: Measurement and modeling. *International Journal of Food Properties* 20, sup2 (Dec. 2017), 1965–1981.

[63] Steven A. Shafer. 1985. Using color to separate reflection components. *Color Research & Application* 10, 4 (1985), 210–218. arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1002/col.5080100409 doi:10.1002/col.5080100409

[64] Fei Shang, Panlong Yang, Dawei Yan, Sijia Zhang, and Xiang-Yang Li. 2024. LiquImager: Fine-grained Liquid Identification and Container Imaging System with COTS WiFi Devices. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 8, 1, Article 15 (March 2024), 29 pages. doi:10.1145/3643509

[65] Fei Shang, Panlong Yang, Yubo Yan, and Xiang Yang Li. 2022. LiqRay: Non-invasive and Fine-grained Liquid Recognition System. In *Proceedings of ACM Conference on Mobile Computing and Networking (MobiCom'22)*. 296–309.

[66] Fei Shang, Panlong Yang, Yubo Yan, and Xiang-Yang Li. 2023. PackquID: In-packet Liquid Identification Using RF Signals. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 6, 4, Article 181 (Jan. 2023), 27 pages. doi:10.1145/3569469

[67] Neha Sharma, Muhammad Shahzaib Waseem, Shahrzad Mirzaei, and Mohamed Hefeeda. 2023. MobiSpectral: Hyperspectral Imaging on Mobile Devices. In *Proceedings of ACM Conference on Mobile Computing and Networking (MobiCom'23) (ACM MobiCom '23).*

[68] Bangjie Sun, Sean Rui Xiang Tan, Zhiwei Ren, Mun Choon Chan, and Jun Han. 2022. Detecting counterfeit liquid food products in a sealed bottle using a smartphone camera. In *Proceedings of ACM Conference on Mobile Systems, Applications and Services (MobiSys'22)*. 42–55.

[69] Xue Sun, Wenwen Deng, Xudong Wei, Dingyi Fang, Baochun Li, and Xiaojiang Chen. 2023. Akte-Liquid: Acoustic-based Liquid Identification with Smartphones. *ACM Transactions on Sensor Networks* 19, 1 (2 2023). doi:10.1145/3551640

[70] Xue Sun and Chao Feng. 2022. Ls-liquid: towards container-irrelevant liquid sensing on smartphones. In *Proceedings of the SIGCOMM '22 Poster and Demo Sessions* (Amsterdam, Netherlands) *(SIGCOMM '22)*. Association for Computing Machinery, New York, NY, USA, 25–27. doi:10.1145/3546037.3546053

[71] Hamid Tabani, Ajay Balasubramaniam, Elahe Arani, and Bahram Zonooz. 2021. Challenges and Obstacles Towards Deploying Deep Learning Models on Mobile Devices. arXiv:2105.02613 [cs.LG] https://arxiv.org/abs/2105.02613

[72] Johannes Taeger, Stefanie Bischoff, Rudolf Hagen, and Kristen Rak. 2021. Utilization of Smartphone Depth Mapping Cameras for App-Based Grading of Facial Movement Disorders: Development and Feasibility Study. *JMIR Mhealth Uhealth* 9, 1 (Jan. 2021), e19346.

[73] Shoji Tominaga, Shogo Nishi, and Ryo Ohtera. 2021. Measurement and Estimation of Spectral Sensitivity Functions for Mobile Phone Cameras. *Sensors* 21, 15 (2021). doi:10.3390/s21154985

[74] Unsoo Ha, Junshan Leng, Alaa Khaddaj, and Fadel Adib. 2020. Food and Liquid Sensing in Practical Environments using RFIDs. In *Proceedings of USENIX Symposium on Networked Systems Design and Implementation (NSDI'20)*.

[75] Felipe Viel, Renato Cotrim Maciel, Laio Oriel Seman, Cesar Albenes Zeferino, Eduardo Augusto Bezerra, and Valderi Reis Quietinho Leithardt. 2023. Hyperspectral Image Classification: An Analysis Employing CNN, LSTM, Transformer, and Attention Mechanism. *IEEE Access* 11 (2023), 24835–24850. doi:10.1109/ACCESS.2023.3255164

[76] Zhu Wang, Yifan Guo, Zhihui Ren, Wenchao Song, Zhuo Sun, Chao Chen, Bin Guo, and Zhiwen Yu. 2024. LiqDetector: Enabling Container-Independent Liquid Detection with mmWave Signals Based on a Dual-Reflection Model. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 7, 4, Article 186 (Jan. 2024), 24 pages. doi:10.1145/3631443

[77] Muhammad Shahzaib Waseem, Neha Sharma, and Mohamed Hefeeda. 2025. RipeTrack: Assessing Fruit Ripeness and Remaining Lifetime using Smartphones. *IEEE Transactions on Mobile Computing* (2025), 1–16.

[78] Matthew R Weir. 2007. Microalbuminuria and Cardiovascular Disease. *Clinical Journal of the American Society of Nephrology* 2, 3 (2007).

[79] Yanni Yang, Yanwen Wang, Jiannong Cao, and Jinlin Chen. 2022. HearLiquid: Nonintrusive Liquid Fraud Detection Using Commodity Acoustic Devices. *IEEE Internet of Things Journal* 9, 15 (2022), 13582–13597. doi:10.1109/JIOT.2022.3144427

[80] Shichao Yue and Dina Katabi. 2019. Liquid testing with your smartphone. In *Proceedings of ACM Conference on Mobile Systems, Applications, and Services (MobiSys'19)*. 275–286.

[81] Mohammad Reza Zarezadeh, Mohammad Aboonajmi, and Mahdi Ghasemi Varnamkhasti. 2021. Fraud detection and quality assessment of olive oil using ultrasound. *Food Science & Nutrition* 9, 1 (2021), 180–189. doi:10.1002/fsn3.1980

[82] John-Lewis Z Zaukuu, Eszter Benes, György Bázár, Zoltán Kovács, and Marietta Fodor. 2022. Agricultural Potentials of Molecular Spectroscopy and Advances for Food Authentication: An Overview. *Processes* 10, 2 (2022).

A Supplementation Materials

This appendix provides more detailed descriptions and results that complement the main paper.

A.1 Feasibility Study: Milk and Urine

We extend our feasibility analysis in §3.1 to two additional liquids, milk and urine, demonstrating the applicability of hyperspectral imaging in diverse contexts relevant to food quality assessment and medical diagnostics. Specifically, we employ the same hyperspectral imaging setup detailed earlier (Specim IQ camera, 204 spectral bands spanning 400–1000 nm, and halogen illumination).

Figure 21 and Figure 22 extend the analysis to two additional domains. In Figure 21, we analyze milk samples with varying fat and lactose contents. Differences in absorption and scattering characteristics across the spectrum reveal measurable variations among samples, especially in the NIR region, where fat and sugar (lactose) contents have a stronger optical influence.

Albuminuria is an early indicator of kidney dysfunction, in which albumin leaks into the urine due to compromised filtration [47]. Figure 22 presents an experiment on urine samples representing three clinical states: (i) healthy (10 mg/100 mL protein), (ii) borderline kidney impairment (15 mg/100 mL protein), and (iii) albuminuria or proteinuria (30 mg/100 mL protein) [15]. Since proteins absorb and scatter light differently than other urinary components, spectral analysis can distinguish these states non-invasively, offering a potential tool for the early detection of renal disease.

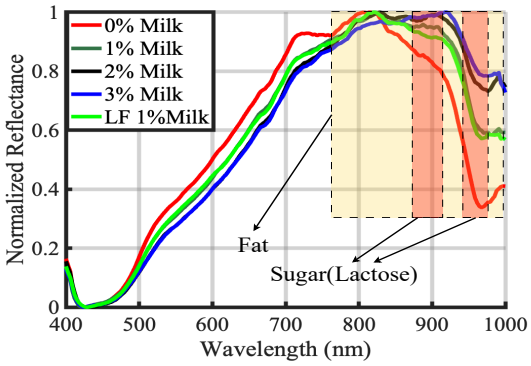

Fig. 21. Analyzing Milk.

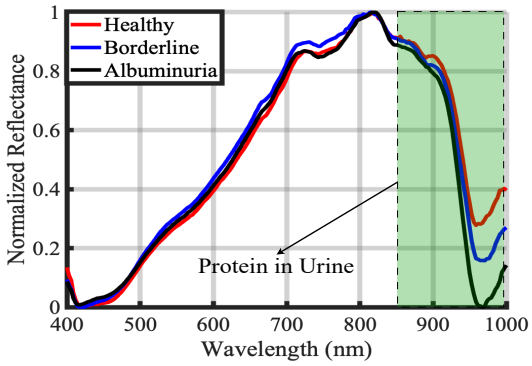

Fig. 22. Screening for Kidney Disease.

A.2 Details of the Reconstruction and Classification Models

Optimization and Training of the Truthful Spectral Reconstruction (TSR) Model. The proposed TSR model has two components. The first one computes the four RGB and NIR bands using Equation (3). This ensures the truthfulness of these bands as they are critical for liquid analysis. The second component uses Equation (4) to estimate the remaining bands in the VNIR spectrum, but it requires using a deep neural network. This neural network is based on the reconstruction model in [67], but with multiple optimizations. The first optimization is adding a loss function relying on a combination of weighted metrics, i.e., MRAE, SAM, and SID, rather than only relying on MRAE. This enables the model to leverage both spectral accuracy and pixel-level accuracy. The loss function is represented by $L = MRAE + 0.1 \times SAM + 0.001 \times SID$. The weights are used to prevent the added losses from overpowering the loss values, which can skew the results. The values are chosen according to [1].

The second optimization is the usage of Multi-head Spectral-wise Attention Blocks (M-SAB). M-SAB were first introduced [10] and later used in [67]. However, we reduce the number of blocks to only two to make the model lightweight. Our evaluation in §5.1 showed that these optimizations improved the reconstruction accuracy and reduced the computation complexity.

The training of the reconstruction model is conducted using a hyperspectral imaging dataset comprising paired hyperspectral images and their corresponding RGB-NIR representations. The hyperspectral data encompasses 68 spectral bands sampled from the visible and near-infrared (VNIR) spectrum at 3 nm intervals, spanning wavelengths from 400 nm to 1000 nm. The dataset is partitioned into training, validation, and test sets to ensure robust model evaluation and prevent overfitting.

The model undergoes training for 50,000 iterations, equivalent to 50 epochs with 1,000 iterations per epoch. Optimization is performed using the Adam optimizer with an initial learning rate of 4×10^{-4}, momentum parameters $\beta_1 = 0.9$ and $\beta_2 = 0.999$, and a weight decay of 10^{-4} to prevent overfitting. A cosine annealing learning rate scheduler is employed to gradually reduce the learning rate to a minimum of 10^{-6}, facilitating convergence to optimal parameters. The training process utilizes a batch size of 20, with input patches of 64×64 pixels extracted using a stride of 64 to ensure comprehensive spatial coverage. Data augmentation techniques are incorporated during training to enhance model generalization capabilities and robustness to variations in the input data.

Model performance is evaluated every 1,000 iterations using the validation dataset, with model checkpoints saved when the total validation loss demonstrates improvement or at regular intervals of 5,000 iterations to preserve the best-performing model states.

Design and Training of the Classification Model. We implemented our classifier as a one-dimensional convolutional neural network (1DCNN), specifically designed to process hyperspectral signatures represented as 68-dimensional vectors. This architecture effectively leverages the sequential nature of spectral data, capturing local correlations between adjacent wavelengths. Furthermore, the 1DCNN architecture has demonstrated both powerful performance and memory efficiency in prior studies [75].

The network architecture consists of three sequential convolutional blocks. Each block comprises a 1D convolutional layer with a kernel size of 3 and padding of 1, preserving the input dimensionality. This is followed by batch normalization, a ReLU activation function, and a max-pooling layer with a stride of 2, which reduces spatial dimensions and introduces translation invariance. The number of convolutional filters progressively increases from 32 in the first block to 64 in the second, and finally to 128 in the third block, allowing the network to learn increasingly complex and abstract spectral features. After the convolutional blocks, the resulting feature maps are flattened and passed through a fully connected dense layer containing 256 hidden units, activated by a ReLU function. Dropout regularization with a dropout rate of 0.3 is applied to this layer to prevent overfitting by reducing co-adaptation among neurons. The final output layer is a fully connected linear layer with units equal to the number of target classes, followed by a softmax activation function, which provides normalized class probability outputs. The model training employs the AdamW optimizer, chosen for its effective decoupling of weight decay from gradient-based parameter updates, thus enhancing L2 regularization and promoting better generalization. The initial learning rate is set at 1×10^{-3}, and a learning rate scheduler based on the `ReduceLROnPlateau` strategy dynamically adjusts the learning rate when validation loss improvement plateaus, ensuring finer convergence in the later training stages.

To address potential class imbalance within the dataset, a weighted cross-entropy loss function is used, with class weights calculated inversely proportional to their frequencies. This approach ensures balanced sensitivity across all classes, preventing bias towards the majority classes and maintaining the model's ability to accurately classify minority classes.

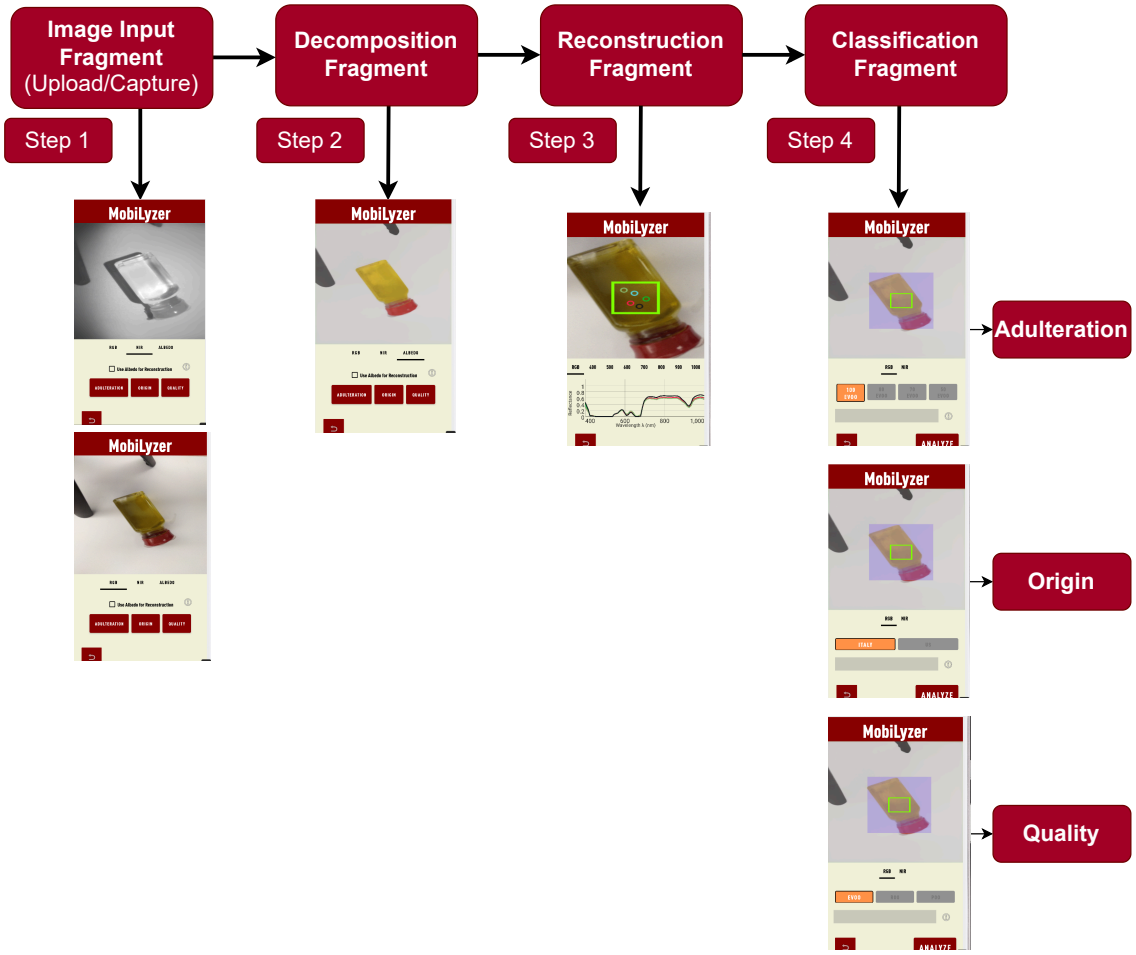

Fig. 23. Overview of the modular app architecture, showing the image input, decomposition, reconstruction, and classification stages. Screenshots from each corresponding fragment are shown in the app interface.

Model training is conducted using stratified 4-fold cross-validation on the training dataset, maintaining class distribution consistency across all folds. Within each fold, the training dataset is partitioned into 75% training and 25% validation subsets. The best-performing model checkpoint is identified based on the minimum validation loss for each fold, and the final evaluation is performed using a fixed 25% hold-out test set that remains completely unseen during training.

Additionally, to optimize model performance, we employ the Optuna hyperparameter optimization framework [3]. Using a Bayesian optimization strategy, specifically the Tree-structured Parzen Estimator (TPE), we systematically explore the hyperparameter space, including learning rate, batch size, and dropout rate. The optimization goal is to maximize cross-validation accuracy while ensuring robust model generalization.

A.3 Design of the MobiLyzer App

MobiLyzer is implemented using a modular architecture consisting of four main components, each implemented as a fragment within the Android framework (see Figure 23). The image input fragment handles RGB+NIR image collection, either through user-uploaded files or real-time capture using the embedded face detection cameras (available by enabling developer options on Android). These image pairs serve as input to the pipeline. The decomposition fragment processes the RGB image to perform intrinsic decomposition and recover the albedo representation §4.2. The reconstruction fragment then uses either the albedo or the original RGB image, depending on user selection, along with the NIR input to perform truthful spectral reconstruction §4.3. Finally, the classification fragment performs one of three tasks: adulteration detection, quality grading, or origin identification, depending on the selected mode .MobiLyzer is implemented in Kotlin [38] and relies on ONNX Runtime [24] as a framework-agnostic tool for deploying deep learning models on mobile devices.

A.4 Intrinsic Decomposition On Mobile Devices

Challenges with Current Intrinsic Decomposition Implementation. Intrinsic decomposition mainly relies on pretrained models, which we covered in the §4.2. Within their core implementation, these models require specific tensor-based operations and NumPy-based functions for input preprocessing or postprocessing at every stage. While these operations are trivial to implement using Python libraries such as NumPy or skimage, they lack direct support in the Android Studio environment.As a result, exporting the pretrained checkpoints directly is not feasible in our case. Reimplementing the preprocessing and postprocessing steps would be required inside Kotlin, often using non-vectorized operations. This can be slow to execute, does not benefit from hardware-accelerated execution kernels, and is also highly error-prone.

Pipeline Re-Implementation. The core insight is that by understanding how PyTorch[56]works under the hood, we can take advantage of its feedforward structure. A feedforward pass in PyTorch directly constructs a computation graph that encodes all the linear and nonlinear layers of the model using supported PyTorch operations. Leveraging this structure, it becomes straightforward to encapsulate respective fragments of our application into modular torch.nn.Module classes, where all intermediate steps and postprocessing operations are encoded as private functions within each class.

Since the official implementation was not originally designed for mobile deployment, many helper utilities were implemented using NumPy, skimage, or other CPU-only Python libraries. These operations, when combined with PyTorch, cannot be traced into the computation graph and are therefore invisible to ONNX export. We carefully reimplemented those operations using PyTorch-friendly counterparts. This allows us to export each component using torch.onnx.export into a platform-agnostic format. Under the hood, ONNX maps the exported computation graph to the corresponding execution table provided by the runtime host—known as the execution provider.

Stage 1 Grayscale Intrinsic Decomposition Official Implementation. Stage 0 uses a simplified Lambertian grayscale intrinsic model of the form $I(\cdot) = R_g^a(\cdot)\, R_g^s(\cdot)$, where $I(\cdot)$ is the linear RGB input, $R_g^a(\cdot)$ is the 3-channel albedo, and $R_g^s(\cdot)$ is the grayscale shading. This process begins by linearizing the input sRGB image. Then, low- and high-resolution ordinal shading maps are estimated to provide global and local constraints, which guide the full intrinsic decomposition network.The prediction is performed in the *inverse shading space*, where shading is modeled as $R^s_{\text{inv}}(\cdot) = \frac{1}{R^s(\cdot)+1}$. To recover the gray scale shading from this representation, the inverse operation $R^s(\cdot) = \left(\frac{1}{\text{clamp}(R^s_{\text{inv}}(\cdot),\epsilon,1.0)}\right) - 1$ is applied. Finally, the albedo is computed from the estimated shading and linear input image using $R_g^a(\cdot) = \frac{I_{lin}(\cdot)}{R_g^s(\cdot)}$.

Stage 1 Grayscale Intrinsic Decomposition Mobile Implementation. We modify Stage 1 implmentation by first encapsulating the entire pipeline within a `torch.nn.Module` class to ensure are pre / post processing steps are encoded within PyTorch's computation graph. All NumPy-based operations are replaced with equivalent PyTorch tensor computations, while `skimage.resize()` calls are replaced with `torch.nn.functional.interpolate()`, using the `antialias=False` because anti aliasing is not supported in ONNX's operator set.

Stage 2 Chroma Estimation Official Implementation. The chroma estimation stage extends the grayscale intrinsic model by relaxing the assumption that the albedo chromaticity matches that of the input image. The goal of this stage is to correct the color shift present in the albedo by transitioning from the grayscale model $I(\cdot) = R^a_g(\cdot)\ R^s_g(\cdot)$ to the RGB diffuse model $\mathrm{I}(\cdot) = \mathrm{R}^a_c(\cdot)\ R^s(\cdot)$, where $R^s(\cdot) = R^s_g(\cdot)R^s_c(\cdot)$. To achieve this, a chroma network is used to predict the shading chromaticity $R^s_c(\cdot)$, given the grayscale albedo $R^a_g(\cdot)$, the input image $I(\cdot)$ transformed into the LUV color space, and the grayscale shading luminance $R^s_g(\cdot)$. The chromaticity prediction is mapped into the $[0, 1]$ space using the transformation $C = \left[\frac{1}{U+1}, \frac{1}{V+1}\right]$. The predicted chromaticity is then concatenated with the shading luminance and transformed back to the RGB space to produce the full RGB colorful shading $R^s(\cdot)$. The corrected albedo is computed as $R^a_c(\cdot) = \frac{\mathbf{I}}{R^s(\cdot)}$. Finally, the output albedo is clipped such that its minimum is approximately 0.001 to avoid overly dark regions, and its 99th percentile is scaled to around 0.75 to prevent overexposure.

Stage 2 Chroma Estimation Mobile Implementation. Similar to Stage 1, we modify the official Stage 2 implementation for mobile deployment by encapsulating the entire pipeline within a `torch.nn.Module` class. We then replace all `skimage` operations with equivalent `torch.nn.functional.interpolate()` calls and substituting NumPy-based preprocessing/postprocessing steps,including RGB to LUV conversion, inversion, and LUV to RGB mapping with PyTorch tensor operations. Since the `torch.quantile` function used for albedo normalization is not supported in ONNX, and directly scaling by the maximum value is sensitive to outliers, we instead implement an approximate quantile function using a `topk`-based approach. Specifically, we compute the minimum value among the top k fraction of pixels where $k = 1 - q = 0.01$ to estimate the 99th percentile and scale the albedo such that this value maps to 0.75.

Stage 3 Albedo Estimation. As described in [13], the chroma estimation phase produces a low-frequency output, and thus only a coarse estimation is required. The chroma network operates at the receptive field resolution and its output is later upsampled to match the input image resolution. However, this coarse estimation can later lead to artifacts in the output color corrected albedo $R^a_c(\cdot)$. To refine this, Stage 3 applies an albedo refinement network that takes as input $R^c_g(\cdot)$, the RGB shading $R^s_c(\cdot)$, and the linear image $I_{lin}(\cdot)$, and outputs the final diffuse albedo $R^a_d(\cdot)$. Given the simplicity of this stage, we do not modify the architecture from the official implementation, except for wrapping the entire stage in a `torch.nn.Module` class to enable ONNX export.

Sanity Check: Validation of our Implementation. To verify the correctness of our ONNX-compatible reimplementation of the §4.2, we follow the same evaluation metrics as in [13]. Specifically, we compute three standard image-level metrics, LMSE, RMSE, and SSIM, between the albedo outputs of our ONNX model and those from the original implementation on the ARAP dataset [9]. As shown in Table Table 5, we obtain albedo predictions that closely match those of the original pipeline, with only subtle discrepancies. These mainly arise from the approximate top-k normalization strategy adopted in Stage 2. Unlike the original implementation, which uses the exact 99th percentile for scaling, our method estimates this quantile using a `topk`-based approximation to ensure ONNX export compatibility. This approximation introduces slight deviations in the computed scaling factor, which in turn causes the albedo predictions, particularly in high-intensity regions, to appear slightly darker than those produced by the official implementation, as shown Figure 24

Table 5. Validation of our optimizations of the intrinsic decomposition model to run on smartphones.

LMSE ↓	RMSE ↓	SSIM ↑
0.01277	0.03844	0.97585

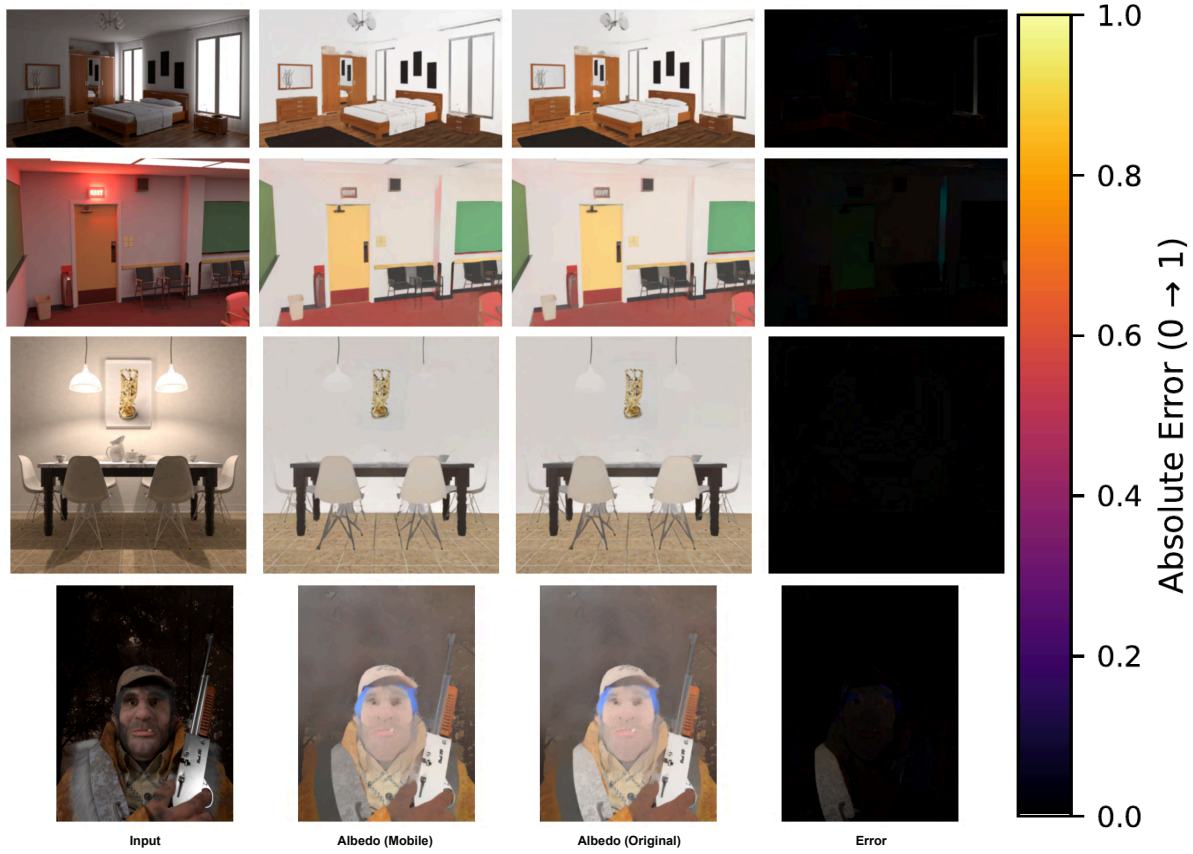

Fig. 24. Qualitative comparison between our mobile-friendly ONNX intrinsic decomposition and the original pipeline.

A.5 Computational Complexity of MobiLyzer

Performance Analysis and Tradeoffs. We identify the primary runtime and memory bottlenecks of deploying MobiLyzer on the OnePlus 8 Pro smartphone using Kotlin's profiler. As shown in Table 6, most of the inference time and memory consumption are concentrated in the intrinsic decomposition §4.2 step.

Quantization Strategy. Quantization is a model optimization technique that improves efficiency by representing weights and activations with lower-precision numerical formats, such as 8-bit integers, instead of 32-bit floating-point values. As noted in the ONNX documentation, *static quantization* can improve both weight and activation efficiency by calibrating on a representative dataset. However, its performance is highly sensitive to the quality of the calibration data. Since our decomposition models are designed for *in-the-wild generalization*, relying on a

Table 6. Inference time and memory requirements of MobiLyzer on OnePlus 8 Pro.

	No Quantization		With Quantization	
	Time (s)	Memory (MB)	Time (s)	Memory (MB)
Intrinsic Decomposition (Stage 1)	2.64	462.25	1.72	118.11
Intrinsic Decomposition (Stage 2)	1.91	398.39	1.05	101.10
Intrinsic Decomposition (Stage 3)	3.82	398.40	2.12	101.08
Intrinsic Decomposition (Total)	8.37	1259.04	4.89	320.29
Spectral Reconstruction	0.67	7.4	0.68	7.4
Classification	0.001	1.3	0.001	1.3
Total (End-to-End)	**9.04**	**1267.74**	**5.57**	**328.99**

Table 7. Quantization impact on albedo reconstruction quality, ONNX model size, and execution time on OnePlus 8 Pro. The best trade-off is highlighted.

Quantized Stages	**Model Size (MB)**	**Time (s)**	**LMSE**	**RMSE**	**SSIM**
None (W/O Quantization)	1259.04	9.04	0.01395	0.03272	0.97917
[1]	914.90	8.12	0.09796	0.09175	0.86873
[2]	961.75	8.18	0.01448	0.03751	0.97891
[3]	961.72	7.34	0.01303	0.03432	0.97105
[1, 2]	617.61	7.26	0.09643	0.09179	0.86905
[1, 3]	617.58	6.42	0.09415	0.09032	0.86500
[2, 3]	**664.43**	**6.48**	**0.01369**	**0.03720**	**0.97146**
[1, 2, 3]	320.29	5.56	0.09509	0.09193	0.86274

limited or biased calibration set risks a significant drop in output quality under constrained testing conditions. In addition, the decomposition models mainly use *MiDaS* CNNs as backbones; thus, poorly calibrated activation ranges can introduce visible artifacts or degrade performance under novel conditions. Instead, we apply *8-bit dynamic quantization*, which only quantizes the weights and performs activation quantization on-the-fly during inference. This strategy yields a substantial reduction in model size from 1.23 GB to 329 MB and nearly halves the inference time, all while maintaining robust generalization across diverse input conditions (Table 6). Given the acceptable end-to-end runtime (~6–11 s) and its robustness to unseen data, *dynamic quantization* is the preferred strategy for MobiLyzer.

Quantization Tradeoffs Across Intrinsic Decomposition Stages. As shown in Table 7, quantizing Stage 1 results in the most noticeable drop in albedo quality, as its outputs serve as input to the remaining two stages. On the other hand, quantizing only Stage 2 and Stage 3 while keeping Stage 1 in full precision achieves significant reduction in the memory footprint without compromising output quality.

A.6 Generalization of MobiLyzer to Unseen Adulterant

The reconstruction model in MobiLyzer is trained for each liquid. We analyze the robustness of this model and its ability to generalize to new liquids not included in the initial training dataset. This is done using transfer learning. We fine-tune the truthful reconstruction model on a small dataset of unseen adulterated Extra Virgin Olive Oil

Table 8. Accuracy of the reconstruction model on Sunflower Oil (not seen during training) before and after transfer learning.

MRAE	RMSE	SAM	SID	SSIM	PSNR
Before Transfer Learning					
0.958	0.059	0.302	0.253	0.695	24.8
After Transfer Learning					
0.304	0.010	0.080	0.025	0.977	39.9

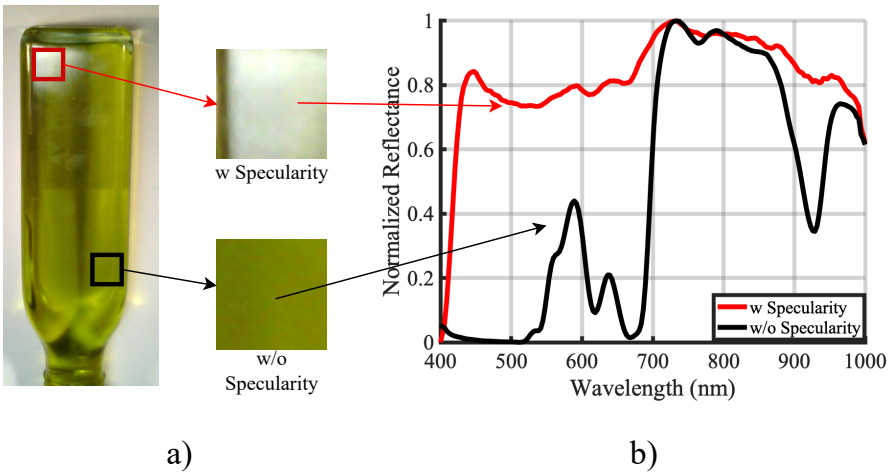

Fig. 25

(EVOO) mixed with Sunflower oil (SO). This dataset comprises liquid mixtures with different percentages: 80% EVOO + 20% SO, 70% EVOO + 30% SO, and 50% EVOO + 50% SO.

Following the common practice in transfer learning, we freeze the majority of the model's layers, allowing only the final layers responsible for output generation to be updated during training. This targeted training is conducted using a limited number of images of the target liquid mixtures; we only use 10 hyperspectral images for each mixture in the fine-tuning. The results of this fine-tuning process on unseen adulterants are presented in Table 8.

As shown in Table 8, applying transfer learning significantly improves the model's performance on the unseen Sunflower oil adulterant compared to its initial performance. This indicates that our model can be effectively fine-tuned with a small amount of data for new, previously unseen liquids and adulterants, thereby substantially increasing its practicality in real-world deployment.

A.7 Impact of Container Interference on Spectral Signatures

We perform an experiment to show how the container can interfere with the captured spectral signatures of liquids. Using the same setup described in §3.1, we capture an extra virgin olive oil (EVOO) sample under a halogen lamp. In the same image, we identify two regions: one affected by specular reflection due to the angle of the light hitting the container, and another with clean reflection directly from the liquid.

Figure 25.a) shows these regions. We select one pixel from the center of each region and plot their spectral signatures in Figure 25.b). The red curve shows the distorted signature from the specular region, while the black curve shows the clean EVOO signature without distortion. This demonstrates that reflections from the container can significantly affect the measured spectral signature.