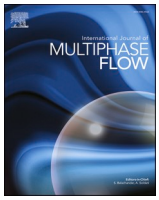
International Journal of Multiphase Flow 194 (2026) 105479



Contents lists available at ScienceDirect

International Journal of Multiphase Flow

journal homepage: www.elsevier.com/locate/ijmulflow



Research Paper



Dense-gas/liquid two-phase flow pattern recognition using convolutional neural networks and transfer learning

July A. Gómez-Camperos [a,b,*], Carlos M. R. Diaz [b], Marlon M. Hernandez-Cely [c], Oscar M. H. Rodriguez [b], Aldo Pardo-Garcia [d]

[a] Mechanical Engineering Department, Universidad Francisco de Paula Santander, Seccional Ocaña, Vía Acolsure, Sede el Algodonal Ocaña, Ocaña 546552, Colombia
[b] Industrial Multiphase Flow Laboratory (LEMI), Mechanical Engineering Department, São Carlos School of Engineering (ESSC), University of São Paulo (USP), Avenida João Dagnone, 1100, Jd. Santa Angelina, São Carlos, SP 13563-120, Brazil
[c] Control and Automation Engineering, Engineering Center, Federal University of Pelotas, Rua Benjamin Constant, N° 989, Porto, Pelotas, RS 96010-020, Brazil
[d] Grupo Automatización y Control (A&C), Universidad de Pamplona, Pamplona 543050, Colombia

ARTICLE INFO

Keywords:
Dense-gas/liquid flow
Two-phase flow
Convolutional neural network (CNN)
Image classification
Transfer learning

ABSTRACT

Accurate identification of dense-gas/liquid two-phase flow patterns is essential in pipeline fluid transport systems and oil well operations. However, conventional methods for classifying these patterns rely mostly on direct visual observation and sensors installed in the system. This study presents an innovative approach for flow pattern recognition, based on modified convolutional neural network (CNN) algorithms and transfer learning based on regularization techniques to achieve an automatic and objective identification of patterns in dense-gas/liquid flows. Experiments were carried out in the Multiphase Flow Experimental Platform of the Industrial Multiphase Flow Laboratory (LEMI). Six flow patterns in horizontal pipes were identified, generating a database of 67,710 images, which are divided into three categories: 60 % are used for model training, 20 % for validation and the remaining 20 % for model evaluation. The proposed model was evaluated on eighteen modified convolutional neural network architectures, and in six test sets. The computational results showed that the proposed model was able to identify the two-phase flow patterns correctly with an accuracy of 1.0 with MobileNet, InceptionV3, InceptionResNetV2 architectures, in the fourth test set, standing out in performance among the six evaluated sets without presenting over-fitting signals. In contrast, the ResNet50, ResNet152V2, MobileNetV3Small, MobileNetV3Large, ResNet101V2, InceptionV4, AlexNet and Linet5 models failed to achieve an accuracy higher than 0.2, obtaining the lowest results in the training sets.

1. Introduction

In the context of the oil industry, gas-liquid flow is present in several applications, including oil well production, pipeline transportation, and offshore operations (Shin et al., 2024; Zou et al., 2021; Wang et al., 2024). An important feature in the specific case of the Brazilian pre-salt oilfields is the presence of a high ratio of dense gases and liquids, transported in subsea pipelines, with different inclinations (Vaz et al., 2021). The notorious lack of data from that complex flow has prevented the development of multiphase for models capable of accurately predicting flow quantities that are required for production monitoring and control (Quintino et al., 2023). There is the need to study dense-gas/liquid flow to better understand its dynamics and optimize operations in these specific environments subjected to critical pressure and temperature conditions.

The hydrodynamics of two-phase flow is determined by a complex interaction between the phases, influenced by different critical parameters that condition the variation and formation of flow patterns (Nnabuife et al., 2023). Among these parameters, the superficial velocities of the liquid and gas are determinant (Luo et al., 2023) by establishing the relative proportion of the phases and their interaction. Similarly, pipe geometry plays a fundamental role (Belt et al., 2024), since characteristics such as diameter, inclination and internal surface roughness significantly affect phase distribution and the development of flow patterns (das Neves et al., 2024; Liu et al., 2021). Additionally, other factors that impact include operating parameters, such as fluctuations in pressure (Yang et al., 2024), temperature, velocity and flow

* Corresponding author at: Mechanical Engineering Department, Universidad Francisco de Paula Santander, Seccional Ocaña, Vía Acolsure, Sede el Algodonal Ocaña, Ocaña 546552, Colombia.
E-mail address: jagomezc@ufpso.edu.co (J.A. Gómez-Camperos).

https://doi.org/10.1016/j.ijmultiphaseflow.2025.105479
Received 30 May 2025; Received in revised form 20 September 2025; Accepted 7 October 2025
Available online 8 October 2025
0301-9322/© 2025 Elsevier Ltd. All rights are reserved, including those for text and data mining, AI training, and similar technologies.

Nomenclature

List of symbols

A	Accuracy
CNN	Convolutional Neural Networks
D	Disperse
DC	Dual-continuous
F1Sc	F1-score
FN	False negatives
FP	False positives (incorrect positive predictions).
Js	Superficial Velocity
P	Pressure
P1	Precision
R	Recall
SL	Slug
SF_6	Sulfur Hexafluoride
ST	Stratified smooth
SW	Stratified Wavy
SW/MI	Stratified Wavy with Mixture at the Interface
T	Temperature
TP	True positives (correct positive predictions).
TN	True Negatives
QCV	Quick-Closing Valve
Val-Loss	Validation of Losses

Subscripts

l	Liquid
g	Gas
abs	absolute

Greek symbol

ρg	Density

Table 1
Studies on the identification of two-phase flow patterns using CNN networks.

Reference	Architectures	Fluid	System
(Nie et al., 2022)	InceptionV3, ResNet50, InceptionResNetV2, MobileNetV2	Methane and tetrafluoromethane.	Condensation flow of methane and tetrafluoromethane in a horizontal circular tube
(Zhang et al., 2023)	VGG16, GoogLeNet	(NaAlg–oil, GaInSn–water)	Microchannels (converging coaxial and vertical coaxial)
(Zhang et al., 2024)	LeNet-5, AlexNet, InceptionV1, ResNet-50 y MobileNetV2.	Gas-solid	Gas-solid fluidized bed, through grid simulations.
(Qiumei et al., 2023)	InceptionV2, InceptionV3 InceptionV4, Resnet50, MobilenetV2 MobilenetV3.	Gas-Water	Pipe and experimental device

rate (Dong et al., 2021), as well as the physical properties of the working fluids (AL-Dogail and Gajbhiye, 2021),which include viscosity, surface tension and density (AL-Dogail and Gajbhiye, 2021).

Transition boundaries between flow patterns, and two-phase flow maps, constitute a relevant challenge in hydrocarbon production and transport processes, which are required for flow control. For instance, they can be used to predict the appearance of unstable flow conditions, such as slug flow (Yan et al., 2024). This regime, in particular, is known to accelerate the fatigue of pipeline components (Arabi et al., 2020; Tandoh et al., 2022), shorten the life of associated equipment (Gonçalves and Mazza, 2024; Onuoha et al., 2022), and cause damage to the pipeline (Mohmmed et al., 2021). In this context, it is imperative to develop and implement advanced and accurate techniques for the identification and monitoring of flow patterns in order to improve operational efficiency and mitigate the risks inherent in the oil industry.

Over time, researchers have worked on the identification of flow patterns under different conditions (Xu et al., 2022). Initial studies focused on the elaboration of flow pattern maps (Yaqub and Pendyala, 2024), which are graphical representations that are used to identify the production envelope associated to an specific flow pattern. After the flow patten is identified, proper correlations can be applied to calculate pressure drop, void fraction, heat transfer (Khan et al., 2024). More recently, advanced instrumentation (Shi et al., 2021), data science and artificial intelligence have been implemented, which has made the identification of two-phase flows more practical, accurate and efficient (Kuang et al., 2024). Several artificial intelligence approaches, including machine learning and neural networks, have been developed to effectively analyze complex two-phase flow dynamics. For example, in Mohammed et al. (2022) the authors focus on gamma ray and electrical capacitance signals combined with an artificial neural network model for void fraction measurement, in Yuan et al. (2019) they used four ultrasonic sensors incorporating two Artificial Neural Networks (ANN) to identify the flow patterns and obtain the gas volumetric fraction (GVF); in Salgado et al. (2021) they use a mathematical algorithm based on an artificial neural network to correlate the energy spectra of X-ray transmission and scattering measurements with volumetric fraction.

Artificial intelligence technology has undergone a remarkable development (Zhao et al., 2024), especially in the area of computer vision based on convolutional neural networks. The integration of image processing with machine learning (ML) algorithms has emerged as a viable strategy for automatic flow pattern recognition, minimizing subjectivity in the analysis. In this context, convolutional neural network (CNN) algorithms stand out as innovative tools for the identification and classification of high-dimensional data, such as images, providing a favorable balance between computational cost and high accuracy (das Neves et al., 2024). In the specific case of dense-gas/liquid flow in pipes, no studies using convolutional neural networks (CNNs) for the identification of flow patterns through digital image processing have been reported in the literature. Some related works are shown in Table 1 that use other types of fluids and other types of systems; for example, in Nie et al. (2022) they used InceptionV3, ResNet50, InceptionResNetV2 and MobileNetV2 architectures to classify two-phase flows in adiabatic and condensation systems in horizontal pipes, achieving an accuracy of 97 %.

In Zhang et al. (2023) they used a GoogLeNet network for flow pattern classification in microchannels, achieving a pressure of more than 0.97 in liquid-liquid and gas-liquid two-phase flows (water--soybean oil, water-lubricating oil and argon-water). In the study described in Qiao et al. (2023) they used a neural network (ResNet50) with dual attention mechanism modules, to develop a gas-liquid two--phase flow pattern identification model, using air and water as fluids, obtaining an accuracy of 0.99, and identified four flow patterns, sparse bubbly flow, dense bubbly flow, slug flow and annular Flow in a vertical pipe. In Zhang et al. (2024) they used LeNet-5, AlexNet, VGG-16, InceptionV1, ResNet-50 and MobileNetV2 models for the identification of flow patterns in a gas-solid fluidized bed, and they used high resolution simulations to obtain the data.

To train convolutional neural network (CNN) models, a large dataset of two-phase flow images is essential. However, the available data on dense gas/liquid two-phase flow in the literature is highly limited. So far the identification of these flow patterns is done through subjective

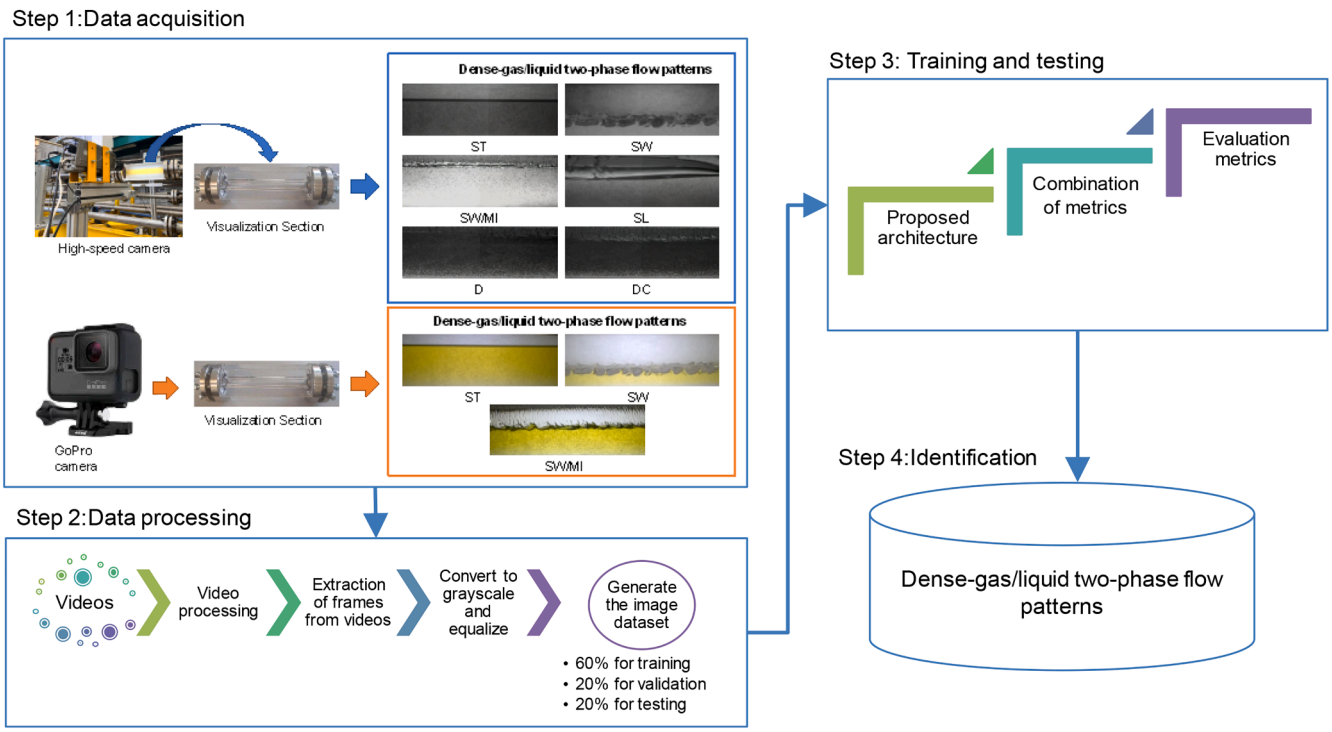


Fig. 1. Schematic diagram of the recognition process for dense-gas/liquid two-phase flow patterns via Convolutional Neural Networks (CNN) and transfer learning techniques.

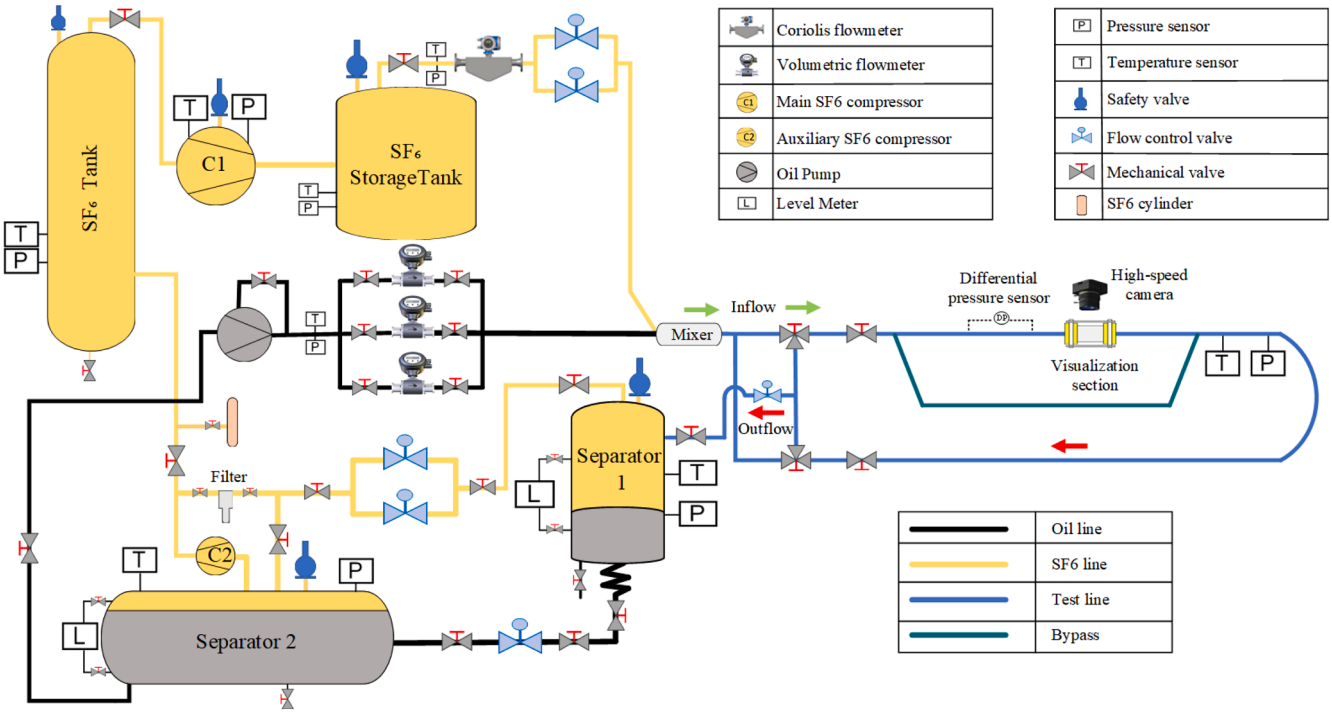


Fig. 2. Schematic diagram of the high-pressure line of the multiphase flow test platform.

methods, as visual observation. We focus on the automatic dense gas/liquid flow pattern identification based on image classification techniques by convolutional neural networks. A complete dataset of flow pattern images from an experiment is used. We propose a novel identification model in the sense that it integrates transfer learning along with a new network and regularization techniques to prevent overfitting. The novelty lies in the application of this specific combination of techniques for the identification of two-phase flow patterns, which has not been done before to the best of the authors' knowledge. Eighteen modified architectures are evaluated on six test sets to demonstrate their effectiveness.

The content of this article is organized as follows. In Section 2 there are the materials and methods, the outline of the experimental platform for data acquisition, data processing, training and testing techniques; In

Section 3 it is provided the experimental results of the dense-gas/liquid two-phase flow patterns, and the results of the six training sets, along with the analysis of each result. Finally, conclusions are summarized in Section 4.

2. Materials and methods

Fig. 1 describes the general steps proposed in this study for the identification of dense gas/liquid flow patterns. The procedure used is structured in four main steps. In the first step, data acquisition is carried out by capturing videos, using a high speed and a GoPro camera, with the objective of recording the flow patterns obtained from the dense gas/liquid experimental platform (details are expanded in Section 2.1). The second step involves data processing, resulting in the generation of an image dataset intended for training, validation and testing. In the third step, training and testing of the proposed architecture is carried out. It is performed using various modified convolutional neural network models, incorporating combinations of regularization techniques and applying Fine Tune. Finally, the accuracy and effectiveness of each model is evaluated by comparing different approaches to identify two-phase flow patterns within the dense-gas/liquid system.

2.1. Data acquisition and processing

The experimental data was collected from the multiphase flow test platform (Fig. 2) installed in the Industrial Multiphase Flow Laboratory (LEMI) at the São Carlos School of Engineering (EESC) of the University of São Paulo (USP).

In this study, sulfur hexafluoride (SF_6) was selected as the working gas phase, considering that in Brazil's pre-salt reservoirs the produced gas contains a high fraction of CO_2 which, when mixed with hydrocarbons and subjected to high pressures and temperatures near or above the critical point, can exhibit dense-gas behavior. Under such conditions, the gaseous phase may reach elevated densities, yielding liquid-to-gas density ratios below 10, which substantially alters two-phase flow hydrodynamics by influencing interfacial entrainment, flow-pattern transitions, and pressure drop. SF_6 achieves high density at moderate pressures, enabling economic and much safer laboratory simulation of dense-gas behavior. For example, whereas carbon dioxide (CO_2) requires pressures exceeding 50 bar to reach densities around 100 kg/m^3, as reported by Carneiro et al. (2015), SF_6 achieves comparable density values at approximately 12 bar. This operational advantage makes SF_6 particularly suitable for dense-gas/liquid flow studies under controlled conditions. In addition, SF_6 exhibits relatively low solubility in mineral oils, minimizing mass transfer effects and ensuring a sharp, well-defined gas–liquid interface throughout the experiments. These characteristics allow mimicking flow behavior from the hydrodynamic point of view as observed in ultra-deepwater production lines such those of the Brazilian pre-salt, without the safety and technical constraints associated with real production fluids.

The liquid and gas storage and separation section consists of four tanks. The SF_6 tank is used for storing the dense gas recovered from the mixture separation tanks. A booster (C1) compresses the SF_6, which flows to a second storage tank. The SF_6 is then mixed with oil, and the mixture flows through the multiphase flow test section. After the test platform, the mixture enters two separation tanks: Separator 1, which receives the gas-liquid mixture and separates most of the dense gas from the oil, and Separator 2, which stores the oil and the remaining SF_6. A second compressor (C2) drives the residual gas back to the SF_6 storage tank.

The multiphase flow test platform is built on an inclinable steel structure equipped with U-shaped stainless-steel pipes 30 m in total length, divided into two 15 m parallel sections. It also includes an acrylic visualization window, pressure and temperature sensors, differential pressure transducers, and video acquisition systems comprising high-speed and conventional cameras.

To control and monitor the injection conditions of the fluids, the dense gas and oil lines are each equipped with dedicated sensors that monitor temperature (T), local pressure (P), and flow rates using turbine and Coriolis flow meters. The gas injection is controlled by defining a mass flow setpoint, regulated by the automatic opening of globe valves installed upstream of the mixer. Similarly, the oil injection is controlled by adjusting the rotational speed (RPM) of the displacement pump according to a predefined volumetric flow setpoint.

The multiphase flow loop incorporates a dedicated gas–liquid separation system that ensures each phase is continuously reinjected practically in its original state. The separation process, which combines the hydrocyclone working principle, controlled depressurization, gravity settling, and filtering, guarantees that the recirculated gas flow is free of oil, while the oil flow pumped back contains no measurable fraction of dissolved or dispersed gas. This operating condition, continuously verified during the experiments, prevents any measurable accumulation of SF_6 in the liquid phase, and ensures that the measured flow rates reliably represent the predefined superficial velocities.

Furthermore, the pressure in the experimental test line is actively maintained at a constant value throughout each test to ensure that the SF_6 remains in a dense phase with a density around 100 kg/m^3. This is achieved via a pressure control system installed in Separator 1, which regulates the discharge of oil and gas toward their respective storage tanks.

For the visualization of flow patterns, two cameras were used: the Olympus i-SPEED 3 and the GoPro Hero 10 5 K. The Olympus i-SPEED 3 is a high-speed camera (HSC) equipped with a custom CMOS sensor with a resolution of 1280 × 1024 pixels and a pixel size of 21 micrometers. It is capable of capturing images at a rate of up to 1 million frames per second (fps) at a resolution of 1280 × 1024 pixels, making it ideal for capturing precise details in high-speed phenomena. On the other hand, the GoPro Hero 10 features a sensor with over 23 MP and the GP2 processor, enabling recordings in 5.3 K resolution at 60 fps and 4 K resolution at up to 120 fps, along with advanced video stabilization (HyperSmooth 4.0) for extreme motion conditions. Both devices were installed in front of a 350 mm long acrylic visualization section. The acquisition rate of the cameras varied between 60 and 3000 frames per second, adjusted according to the specific flow pattern. Fig. 8 details the acquisition rate for each flow pattern.

The video processing was implemented in Python 3.12.5 using a script that leverages OpenCV (version 4.10) and scikit-image (version 0.22.0) for image processing, along with the os module for file handling. The workflow consisted of three main phases:

In the first spatial preparation phase, each video was precisely cropped by removing 600 pixels from the top and 550 pixels from the bottom while preserving the original parameters such as frame rate and horizontal resolution. This was achieved using OpenCV's cv2.VideoCapture and cv2.VideoWriter functions. Next, during the extraction and classification stage, 60 frames per second were systematically selected from each video and organized into specific directories according to their corresponding category. Crucially, human subjectivity in flow pattern labeling was minimized through a rigorous protocol: a team of multiphase flow experts from the LEMI laboratory performed annotations by correlating visual observations with quantitative parameters including pressure differential (ΔP), ensuring physically-grounded pattern identification. Each image was assigned a standardized name following the frame_000001.jpg format, along with the flow pattern name, to ensure data handling consistency. Finally, in the image conditioning phase, several optimization processes were applied: conversion to grayscale (using cv2.COLOR_BGR2GRAY), normalization to the [0, 1] range, contrast enhancement via adaptive histogram equalization (CLAHE with an 8 × 8 window and a limit of 2.5), and resizing to 256 × 256 pixels using bicubic interpolation.

Once the frames were extracted and preprocessed, the dataset was split into training (60 %), validation (20 %), and test (20 %) sets. This division was performed using scikit-learn's train_test_split function,

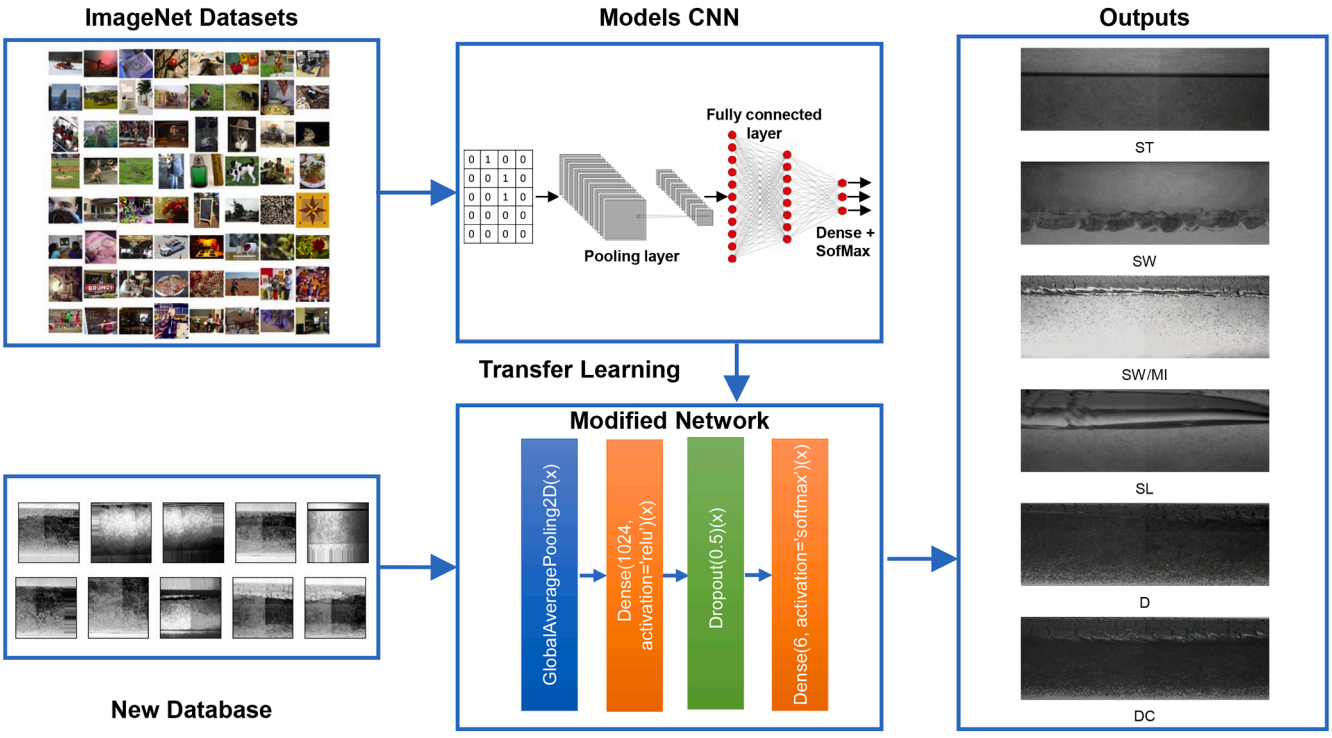


Fig. 3. First Approach, CNN of transfer learning.

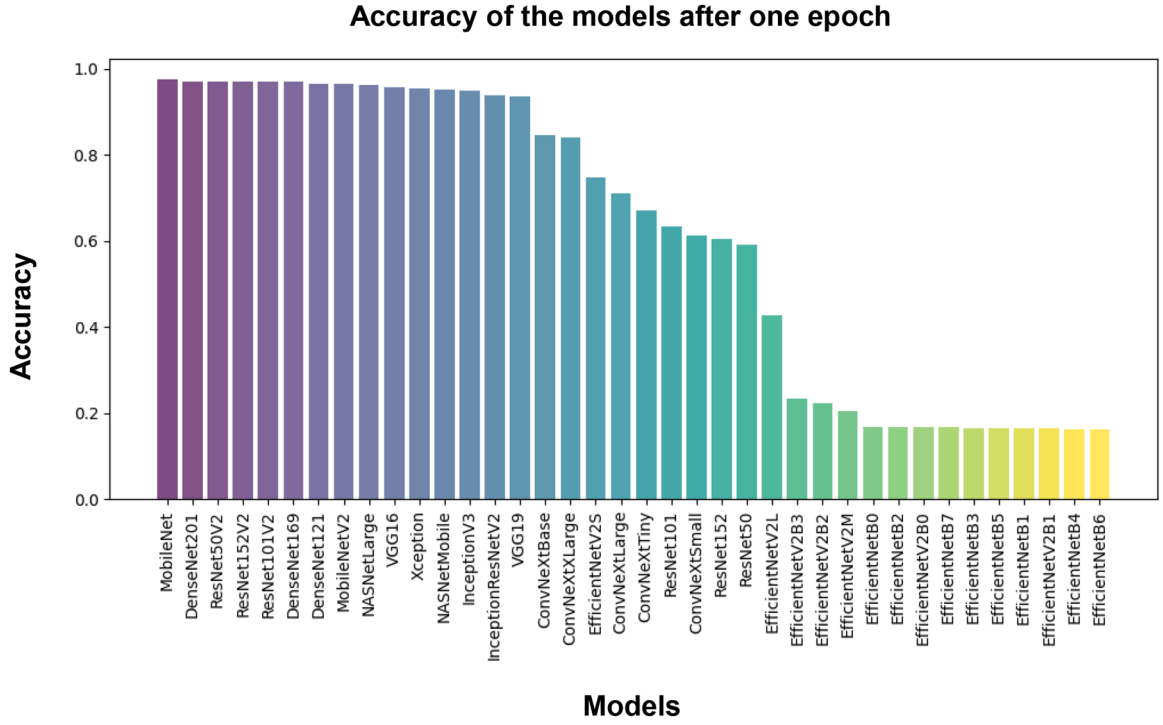


Fig. 4. Accuracy graph after completing a training period.

ensuring a stratified and reproducible distribution with a fixed random seed (random_state = 42). Each subset was stored in separate directories (train, validation, test), maintaining the class folder structure to facilitate their use in machine learning model training. This approach ensured robust evaluation and prevented data leakage between the different partitions

2.2. *Training, validation and testing phases*

For training the convolutional neural network (CNN) models, the Visual Studio Code interpreter was used alongside the open-source libraries TensorFlow, Keras, and Matplotlib. TensorFlow (a comprehensive machine learning platform) provides the foundation for distributed computing operations and graph execution. Keras version 3.4.1 offers a

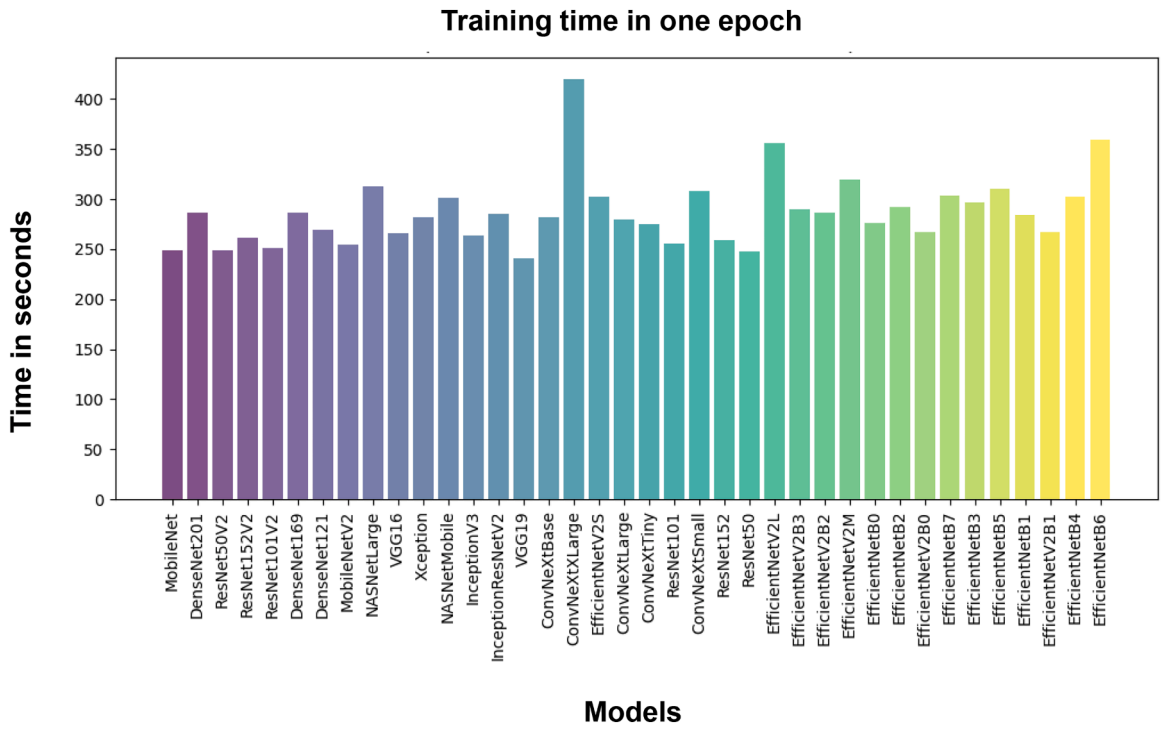


Fig. 5. Training time graph after one epoch.

Table 2
Models trained with less time and higher learning rate.

	References	Model	Train_accuracy	Val_accuracy	Training time (sec)
1	(Howard et al., 2017)	MobileNet	0.975	0.995	248.94
2	(Huang et al., 2017)	DenseNet201	0.971	0.997	286.9
3	(He et al., 2016)	ResNet50V2	0.971	0.992	248.73
4	(He et al., 2016)	ResNet152V2	0.971	0.991	260.95
5	(He et al., 2016)	ResNet101V2	0.970	0.989	250.96
6	(Huang et al., 2017)	DenseNet169	0.970	0.995	286.32
7	(Huang et al., 2017)	DenseNet121	0.965	0.992	268.81
8	(Sandler et al., 2017)	MobileNetV2	0.964	0.992	254.0
9	(Zoph et al., 2018)	NASNetLarge	0.962	0.988	312.74
10	(Simonyan and Zisserman, 2015)	VGG16	0.958	0.985	266.12

high-level API for building neural network architectures with predefined modules, while Matplotlib enables the visualization of metrics and results through 2D plots. This workflow was implemented in Python version 3.12.5 with NumPy version 1.26.4 support for efficient numerical operations.

The employed architecture is based on pretrained models adapted through transfer learning, adding a small network as shown in Figs. 3 and 4. The models were compiled with the Adam optimizer (initial learning rate = 0.001), categorical_crossentropy loss function, and accuracy metric. Training was conducted for up to 100 epochs, implementing early stopping (patience = 5 epochs) and dynamic learning rate reduction (ReduceLROnPlateau with factor = 0.5 and patience = 5), using a batch size of 32. This configuration ensured a balance between generalization capability and efficiency.

The equipment used to execute the work has the following specifications: CPU: 13th Gen Intel(R) Core (TM) i9-13900 K 3.00 GHz, RAM: 32.0 GB (31.8 GB usable), 64-bit operating system, x64 processor and GPU: NVIDIA GeForce RTX 4070.

2.2.1. Proposed CNN model for classification of dense-gas/liquid two-phase flow images

First Approach: in Fig. 3, a schematic view of the first approach can be observed, which is based on Convolutional Neural Networks (CNN) with transfer learning. This approach is grounded in two key stages: Initially, it leverages knowledge from CNN architectures pre-trained on large datasets such as ImageNet, partially freezing the initial layers to preserve their parameters and avoid modifications during training. In the second stage, the original architecture is modified by replacing the fully connected (FC) layers with an adaptive block, and the resulting network is trained on a new, specific database of dense gas/liquid two-phase flow patterns. This block includes a Global Average Pooling 2D layer (to reduce dimensionality while retaining key features), dense layers with ReLU activation (nonlinear combination of attributes), a Dropout layer (overfitting control), and a final softmax layer (classification).

The model, fine-tuned through this approach, successfully identifies six distinct flow patterns (ST, SW, SL, SW/MI, D, and DC). Its effectiveness stems from integrating knowledge acquired via transfer learning from ImageNet, which optimizes resources by eliminating the need to train from scratch, and adapting to the specific distribution of two-phase flow data.

For the selection of the models to be implemented, a pre-evaluation of the performance of the thirty-eight pre-trained models available in

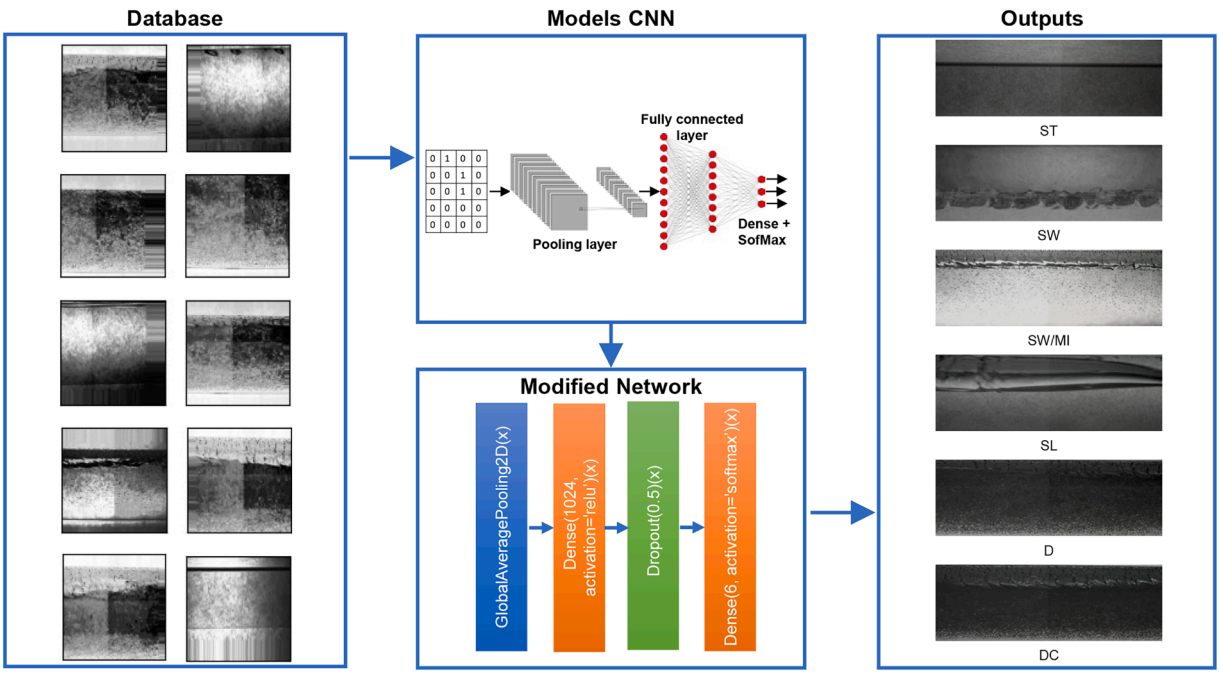


Fig. 6. Second approach, CNN.

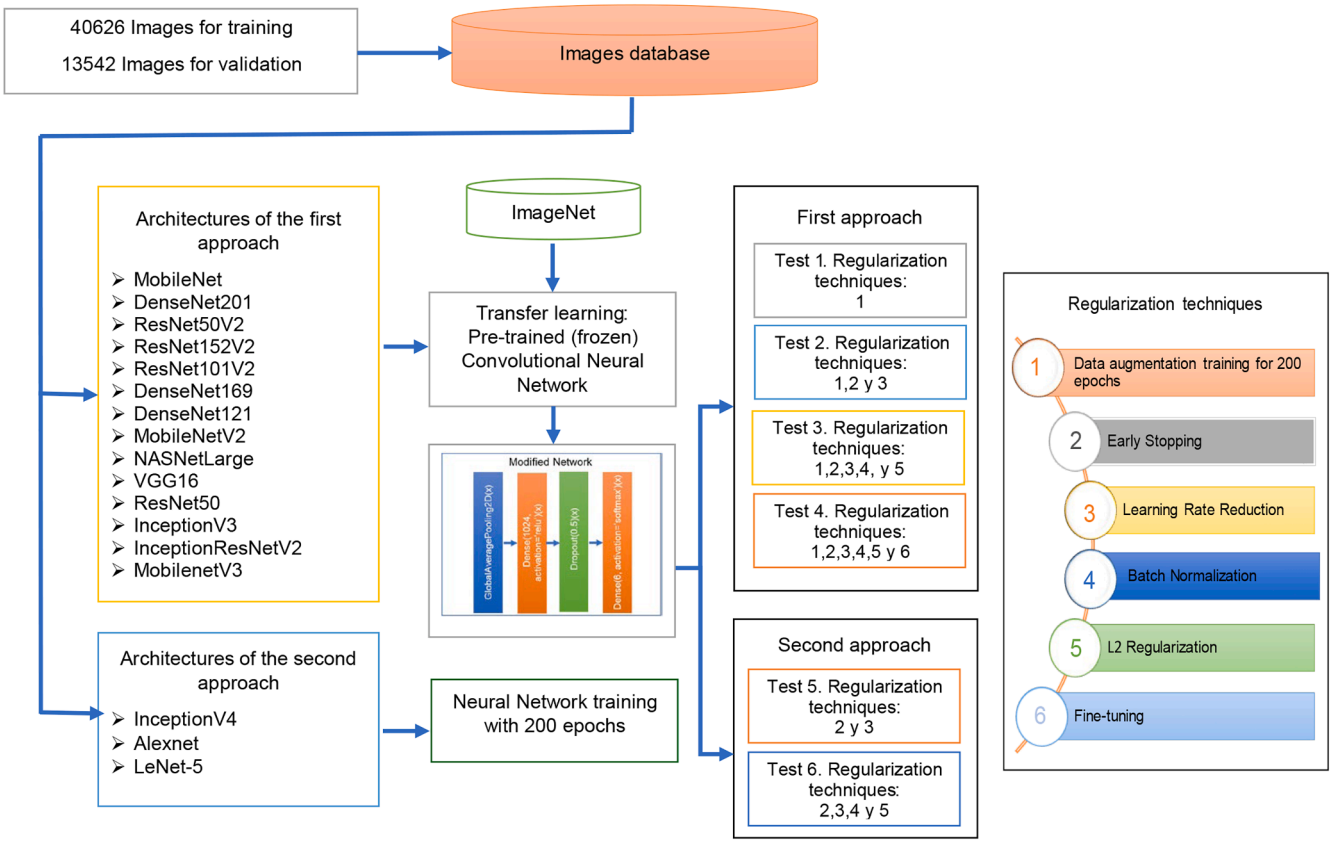


Fig. 7. Test set diagram.

Keras was carried out. This evaluation consisted of analyzing the specific performance of each model using a dataset of dense-gas/oil flow images. During training, the best performing models were identified, considering key metrics such as training accuracy, validation accuracy (Val accuracy) and training time. The detailed results of this pre-evaluation are shown in Figs. 4 and 5.

Analyzing the results of the graphs presented in Figs. 4 and 5, it can be observed that after one training period, the ten models that obtain the best results in comparison of accuracy, accuracy validation and training time are detailed in Table 2. Among them, the MobileNet model stands

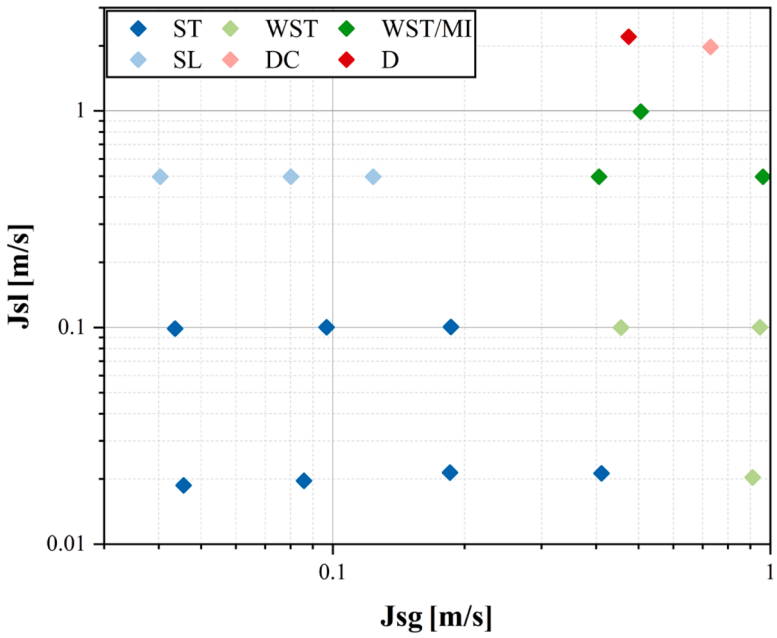


Fig. 8. Experimental flow pattern map at 0°.

out by reaching an accuracy of 0.975 after only one training period. The network configurations are documented in Table 2 through references specifying each network's parameters.

In addition to the ten proposed models, four additional transfer learning models were selected, which were proposed by the works described in Table 2, which are: MobileNetV2, MobilenetV3, ResNet-50, InceptionResNetV2.

Second Approach: in Fig. 6, it is illustrated a flow pattern identification system based on Convolutional Neural Networks (CNN). The system uses a dataset of representative images of various flow patterns as input. The analysis of these patterns is carried out using two CNN architectures: a base model and a modified network. In the base model, pre-existing architectures such as InceptionV4, AlexNet, and LeNet-5 were implemented and evaluated. The modified network is distinguished by the incorporation, in the last layer of the base model, of a network composed of a GlobalAveragePooling2D layer, a Dense layer with a ReLU activation function, a Dropout layer (with a rate of 0.5), and a final Dense layer with a Softmax activation function for classification. The outputs of the system are classified images, representing different flow patterns.

2.3. Regularization techniques and evaluation metrics of the proposed model for flow pattern identification

2.3.1. Regularization techniques

This study employed five techniques for training the two proposed approaches to optimize the model's performance and prevent overfitting. These techniques were evaluated using six test sets, as illustrated in Fig. 7. The Early Stopping technique (Miseta et al., 2024) was employed to halt training when stagnation or deterioration in the model's performance on the validation set was detected, thereby preventing overfitting and enhancing generalization capability. On the other hand, data augmentation allowed for the expansion and diversification of the training set through the application of transformations that preserve the original labels (Mumuni and Mumuni, 2024). Similarly, learning rate reduction on plateau enabled dynamic adjustment of the learning rate in cases where no improvements in the model's performance were observed, thus facilitating convergence toward an optimal solution (Nakamura et al., 2021). Batch normalization was implemented to stabilize the learning process and improve training efficiency by reducing internal covariate shift. Finally, the Fine-Tuning technique was applied, which involves adapting a pre-trained model from a related task to optimize its performance on the specific task, maximizing the utilization of acquired knowledge and accelerating the training process (Yuan et al., 2019).

2.3.2. Evaluation metrics

The following metrics were used: precision, sensitivity (also known as recall), accuracy and F1 score.

Precision: it measures the proportion of correct predictions among all positive predictions made by the model, Eq. (1).

$$Precision = \frac{TP}{TP + FP} \tag{1}$$

Sensitivity: it evaluates the ability of the model to correctly identify all real positive instances, Eq. (2).

$$\text{Recall (Sensitivity)} = \frac{TP}{TP + FN} \tag{2}$$

Accuracy: it measures the proportion of correct predictions (positive and negative) over the total predictions made, Eq. (3).

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{3}$$

F1 Score: it is the harmonic mean of accuracy and sensitivity, useful in cases where there is an imbalance between classes, Eq. (4).

$$\text{F1Sc} = \frac{2TP}{2TP + FP + FN} \tag{4}$$

Specificity: Measures the ability of the model to correctly identify true negatives (that is, when the class truly does not belong to the positive category), Eq. (5).

$$\text{Specificity} = \frac{TN}{TN + FP} \tag{5}$$

It is important to note that these metrics are derived from the generation of the confusion matrix for each model.

2.4. Test sets

Fig. 7 illustrates six test sets, showcasing the two proposed

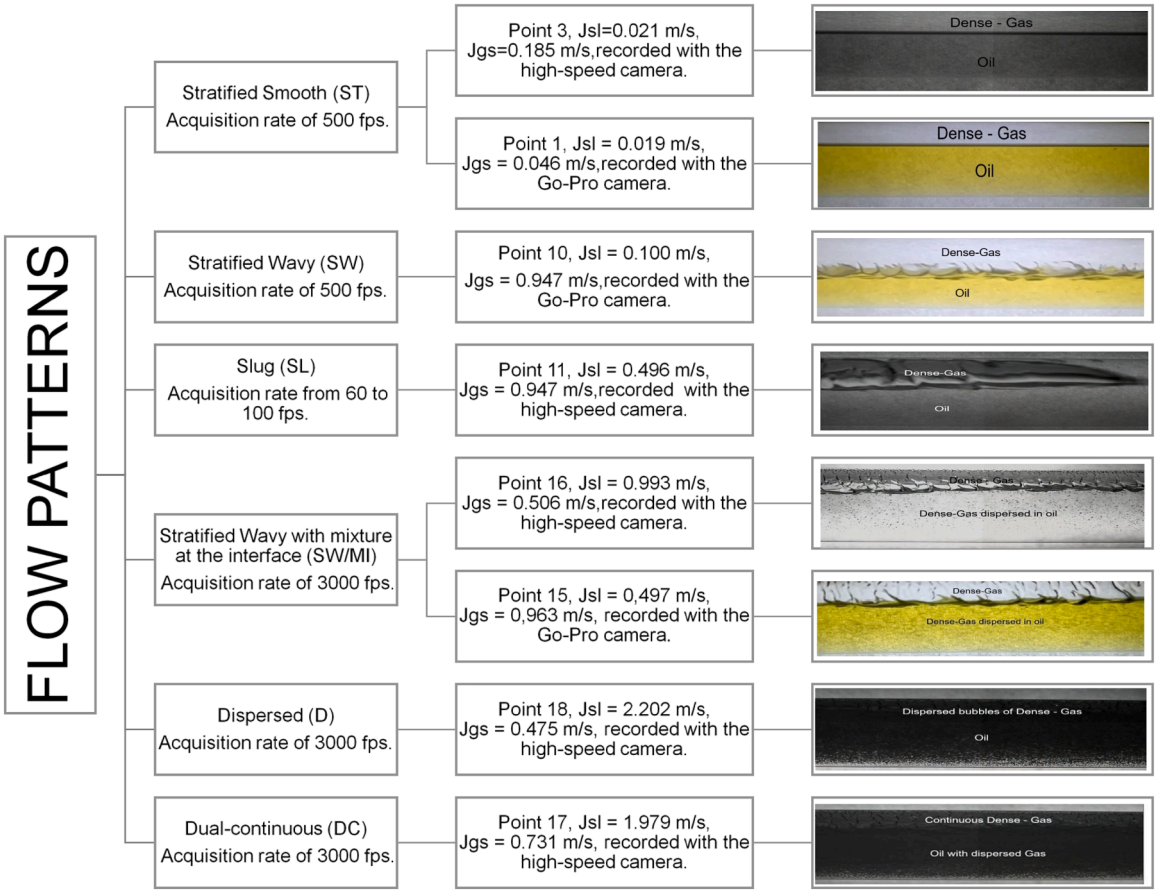


Fig. 9. Flow patterns observed in the experiments.

Table 3
Experimental conditions and observed flow patterns.

Point	P_{abs} [bar]	ρ_g [kg/m^3]	T [°C]	J_{sl} [m/s]	J_{sg} [m/s]	Flow pattern [-]
1	13.7	102.0	21.7	0.019	0.046	ST
2	14.1	104.5	23.5	0.020	0.086	ST
3	14.1	103.1	25.5	0.021	0.185	ST
4	13.9	102.1	25.1	0.021	0.411	ST
5	14.1	105.6	22.4	0.020	0.911	SW
6	14.5	105.2	27.5	0.099	0.044	ST
7	14.1	102.9	24.1	0.100	0.097	ST
8	13.7	98.1	25.5	0.101	0.186	ST
9	13.6	98.8	25.6	0.100	0.456	SW
10	13.8	101.8	23.9	0.100	0.947	SW
11	14.2	106.6	22.4	0.496	0.040	SL
12	13.8	104.4	20.9	0.496	0.080	SL
13	13.7	104.6	19.6	0.496	0.124	SL
14	14.0	104.7	23.3	0.496	0.406	SW/MI
15	13.9	103.1	24.4	0.497	0.963	SW/MI
16	13.7	104.7	19.0	0.993	0.506	SW/MI
17	14.1	103.7	24.4	1.9	0.731	DC
18	13.8	104.4	19.9	2.2	0.475	D

approaches for the CNN-based Proposed Model. For the first approach, four test sets (Test 1, 2, 3, and 4) were created using the ten models evaluated in Table 2, selected for their shorter training time and higher learning rate. Additionally, leveraging prior research, the Inception-ResNetV2, ResNet50, MobileNetV2, MobileNetV, and MobileNetV3 architectures were assessed. Transfer learning techniques were applied to these networks, along with a modified network incorporating regularization strategies and fine-tuning. The second approach comprises two test sets (Test 5 and 6), where CNN models such as InceptionV4, AlexNet, and LeNet-5 were evaluated using regularization techniques.

Regarding the InceptionV1 (also known as GoogLeNet) and InceptionV2 networks, they were excluded from training as they were replaced by InceptionV3, which improved both computational efficiency and model performance.

3. Results and discussions

3.1. Experimental results of flow patterns

During the experiments, eighteen experimental points were obtained with the range of oil superficial velocities from 0.10 m/s to 2.10 m/s and SF_6 velocities from 0.10 m/s to 0.91 m/s, as shown in Fig. 8, and the gas density around 100 kg/m^3. From the eighteen experimental points, six different flow patterns were observed: Stratified Smooth (ST), Stratified Wavy (SW), Slug (SL), Stratified Wavy with mixture at the interface (SW/MI), Dispersed (D), Dual-continuous (DC); these patterns can be observed in Fig. 9. As shown in Table 3, a detailed summary of the experimental points is presented, including the absolute pressures (P_{abs}), gas density (ρ_g), temperatures (T), and superficial velocities of both the liquid (Jsl) and the gas (Jsg), along with the flow pattern observed under each condition. These experimental points were distributed over a temperature range from 19.0 °C to 27.5 °C and pressures ranging from 13.6 bar to 14.5 bar, with variations within this range of approximately 0.85 bar.

Stratified Smooth (ST) flow pattern is characterized by phase segregation due to gravity; the heavier phase (oil) flows at the bottom of the pipe, and the gas (SF_6) flows at the top. The interface between the phases does not exhibit interfacial waves. Stratified Wavy (SW) flow pattern is similar to ST, but the interface between the phases exhibits stable interfacial waves, and slight mixing can also observed at the

Table 4
Results of the architectures in Test 1.

Models	Test Set				Val Loss	Val MSE	Val Error	Epochs
	A	P1	R	F1Sc				
Mobilinet	1.0	1.0	1.0	1.0	0.001179	0.00	0.0005	200
MobilinetV2	1.0	1.0	1.0	1.0	0.001208	0.0001	0.0024	200
NasNeLarge	0.99	0.99	0.99	0.99	0.004643	0.0003	0.0027	200
InceptionV3	0.99	0.99	0.99	0.99	0.006033	0.0003	0.0045	200
DenseNet201	0.98	0.98	0.98	0.98	0.000624	0.0002	0.003	200
ResNet50V2	0.97	0.98	0.97	0.97	0.003423	0.0006	0.0046	200
VGG16	0.97	0.97	0.97	0.97	0.000469	0.0008	0.0082	200
DenseNet121	0.97	0.97	0.97	0.97	0.001915	0,0009	0.0074	200
InceptionResnetV2	0.97	0.97	0.97	0.97	0.005158	0.001	0.0109	200
DenseNet169	0.90	0.93	0.90	0.90	0.000378	0.1904	0.2279	200
MobileNetV3Small	0.55	0.50	0.55	0.50	0.056166	0.5519	0.4411	200
MobileNetV3Large	0.45	0.72	0.45	0.45	0.003160	0.4554	0.3904	200
ResNet50	0.26	0.31	0.26	0.16	0.023645	0.9271	0,8829	200
ResNet152V2	0.17	0.03	0.17	0.05	0.002966	0.9616	0.9221	200
ResNet101V2	0.07	0.02	0.07	0.03	0.002761	0.9904	0.9718	200

Table 5
Results of the architectures in Test 2.

Models	Test Set				Val Loss	Val MSE	Val ER	Epochs
	A	P1	R	F1Sc				
MobileNet	1.0	1.0	1.0	1.0	0.000084	0.00	0.0001	17
InceptionResnetV2	0.98	0.98	0.98	0.98	0.000976	0.0002	0.0003	19
InceptionV3	0.99	0.99	0.99	0.99	0.001373	0.0003	0.0005	19
NasNeLarge	0.98	0.98	0.98	0.98	0.002151	0.0003	0.0004	16
DenseNet201	0.96	0.96	0.96	0.96	0.000283	0.0005	0.0007	15
DenseNet169	0.93	0.95	0.93	0.92	0.003770	0.0006	0.0010	14
DenseNet121	0.91	0.93	0.91	0.90	0.000016	0.1945	0.2202	11
MobilinetV2	0.88	0.93	0.88	0.86	0.000140	0.2445	0.3802	16
MobileNetV3Small	0.58	0.52	0.58	0.52	0.046166	0.5892	0.6023	44
MobileNetV3Large	0.49	0.81	0.49	0.49	0.003160	0.6216	0.6532	20
VGG16	0.42	0.57	0.42	0.34	0.000003	0.6512	0.6732	28
ResNet50V2	0.19	0.14	0.19	0.09	0.001714	0.9283	0.9342	20
ResNet152V2	0.17	0.03	0.17	0.05	0.000003	0.9787	0.9834	11
ResNet101V2	0.17	0.03	0.17	0.05	0.000013	0.9798	0.9884	12
ResNet50	0.17	0.03	0.17	0.05	0.005936	0.9802	0.9724	17

interface. Slug (SL) flow pattern is characterized by alternating flow of liquid and elongated gas bubbles; the liquid slug fills the entire cross section of the pipe and is separated by gas pockets at the top of the pipe and a liquid layer at the bottom of the pipe. The Stratified Wavy with mixture at the interface (SW/MI) flow pattern is similar to SW, but much more mixing is observed, specially of gas bubbles dispersed in the liquid. Dispersed (D) flow is characterized by a continuous liquid phase with dispersed gas bubbles. There is a higher concentration of bubbles at the top of the pipe due to gravity. Dual-continuous (DC) flow pattern is characterized by the maintenance of gas and liquid continuity. However, a very strong dispersion is observed in both phases. The gas phase is concentrated at the top of the pipe and the liquid phase at the bottom.

3.2. Results and comparison of the tests

In Test 1, the transfer learning technique was applied, and four layers were added to the model, as shown in Fig. 7. In addition, the data augmentation technique was used, training each model for 200 epochs. The models were trained with 40,626 images of 6 classes of two-phase flow patterns: ST, SW, SL, SW/MI, D, and DC (Fig. 8).

Table 4 presents the results of fifteen neural network models evaluated on Test1, considering key metrics such as accuracy (A), precision (P1), recall (R), F1-Score (F1Sc), validation loss (Val-Loss), mean squared error (MSE), error, and training epochs (200 in all cases, using the Adam optimizer). The MobileNet and MobileNetV2 models stand out with perfect performance (1.0 in A, P1, R, and F1Sc), alongside minimal validation losses (0.001) and near-zero errors, demonstrating excellent generalization capability. Architectures like NASNetLarge and InceptionV3 also show high performance (0.99 across all metrics), albeit with a slightly higher MSE. Meanwhile, DenseNet201, VGG16, and ResNet50V2 maintain solid performance (F1-Score 0.97–0.98), whereas DenseNet169 exhibits a notable decline (F1-Score = 0.90). In contrast, the MobileNetV3Small, MobileNetV3Large, ResNet50, ResNet152V2, and ResNet101V2 models achieved significantly inferior results (F1-Score between 0.03 and 0.55), with high validation errors (>0.92), suggesting limitations in their adaptation to the evaluated task. It is worth noting that for data augmentation, Keras ImageDataGenerator tool (Shi et al., 2021) was used, configured with: Image rescaling 20 % horizontal/vertical shifts, 20 % zoom, Random cropping (20 % probability), Pixel filling via the "nearest" method.

In Test 2, the early stopping technique, which is used to stop training a model when no improvement is observed in a validation metric, thus avoiding overfitting, and the plateau learning rate reduction technique are used. This technique adjusts the optimizer's learning rate when the validation metric does not improve during a specific number of epochs, helping to improve the convergence of the model. These two techniques are used together with the transfer learning technique, and four layers were added to the model, as shown in Fig. 7. In addition, the data augmentation technique was used, training each model for 200 epochs.

Table 5 presents the results obtained by the models in the second experiment. The combination of early stopping and adaptive learning rate reduction in Test 2 demonstrates significant advantages in computational efficiency and generalization, though it reveals critical limitations in less robust architectures. These techniques successfully

Table 6
Results of the architectures in Test 3.

Models	Test Set				Val Loss	Val MSE	Val ER	Epochs
	A	P1	R	F1Sc				
MobileNet	1.0	1.0	1.0	1.0	0.037	0.0000	0.0005	11
InceptionV3	1.0	1.0	1.0	1.0	0.062	0.0008	0.0024	12
InceptionResnetV2	0.98	0.98	0.98	0.98	0.092	0.0008	0.0027	14
DenseNet169	0.96	0.96	0.96	0.96	0.023	0.0019	0.0045	16
DenseNet201	0.96	0.97	0.96	0.96	0.025	0.0017	0.003	16
DenseNet121	0.94	0.95	0.94	0.94	0.023	0.0023	0.0046	17
NasNeLarge	0.92	0.94	0.92	0.92	0.047	0.0053	0.0082	8
MobilinetV2	0.89	0.93	0.89	0.88	0.026	0.2623	0.3672	13
MobileNetV3Small	0.63	0.66	0.63	0.57	21.583	0.6231	0.4456	8
MobileNetV3Large	0.60	0.67	0.60	0.58	5.464	0.6423	0.4762	14
VGG16	0.51	0.59	0.51	0.46	0.022	0.7867	0.702	14
ResNet50	0.17	0.04	0.17	0.05	7.107	0.9787	0.9834	16
ResNet101V2	0.17	0.03	0.17	0.05	0.036	0.9798	0.9884	18
ResNet50V2	0.17	0.03	0.17	0.05	0.027	0.9802	0.9724	16
ResNet152V2	0.17	0.03	0.17	0.05	0.034	0.9787	0.9834	12

Table 7
Results of the architectures in Test 4.

Models	Test Set				Val Loss	Val MSE	Val ER	Epochs
	A	P1	R	F1Sc				
MobileNet	1.0	1.0	1.0	1.0	0.019	0.0000	0.0002	26
InceptionV3	1.0	1.0	1.0	1.0	0.004	0.0006	0.0024	27
InceptionResnetV2	1.0	1.0	1.0	1.0	0.032	0.0008	0.0027	18
DenseNet201	0.30	0.11	0.30	0.15	0.003	0.8019	0.8045	30
DenseNet169	0.62	0.72	0.62	0.55	0.001	0.6017	0.6003	26
DenseNet121	0.63	0.86	0.63	0.60	0.001	0.6023	0.6046	26
NasNeLarge	0.97	0.97	0.97	0.97	0.008	0.0053	0.0082	31
MobilinetV2	0.99	0.99	0.99	0.99	0.011	0.2623	0.3672	30
MobileNetV3Small	0.35	0.67	0.35	0.25	0.109	0.6231	0.4456	20
MobileNetV3Large	0.49	0.81	0.49	0.49	0.098	0.6423	0.4762	20
VGG16	0.46	0.46	0.46	0.32	0.026	0.7267	0.692	28
ResNet50	0.18	0.10	0.18	0.10	0.246	0.9287	0.9634	12
ResNet101V2	0.17	0.03	0.17	0.05	0.005	0.9798	0.9884	43
ResNet152V2	0.17	0.03	0.17	0.05	0.004	0.9802	0.9724	35
ResNet50V2	0.17	0.03	0.17	0.05	0.004	0.9787	0.9834	31

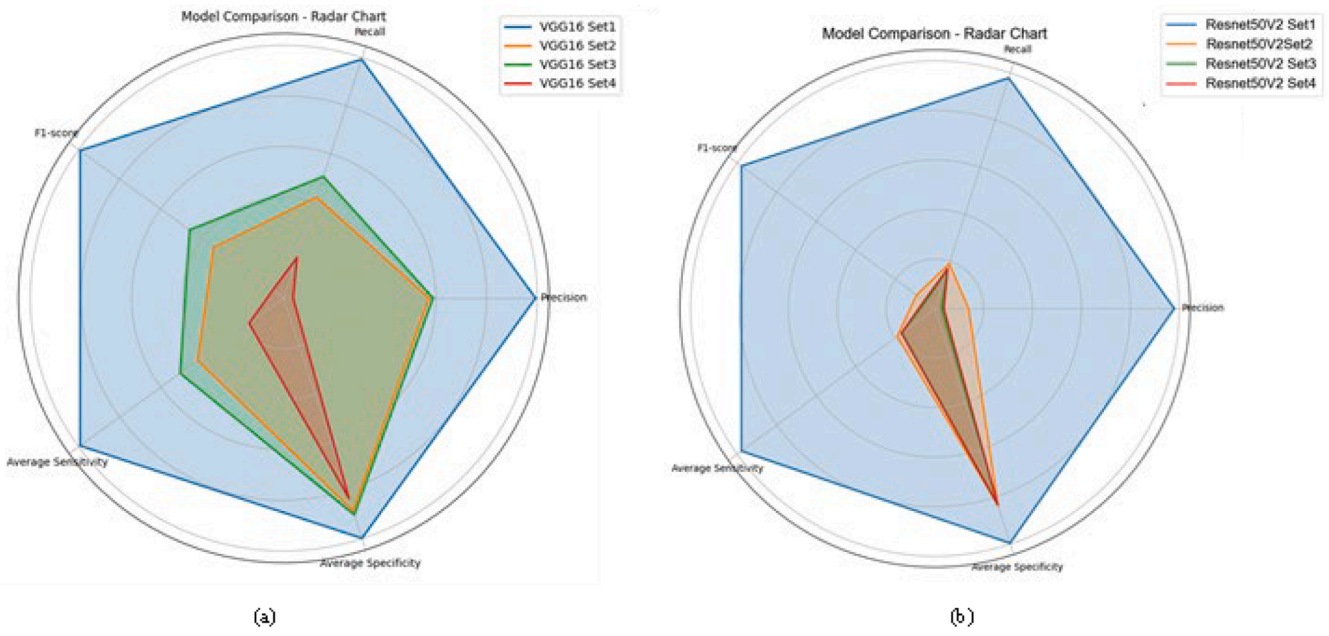


Fig. 10. Radar graph for comparing the four network tests (a) VGG16 and (b) ResNet50V2.

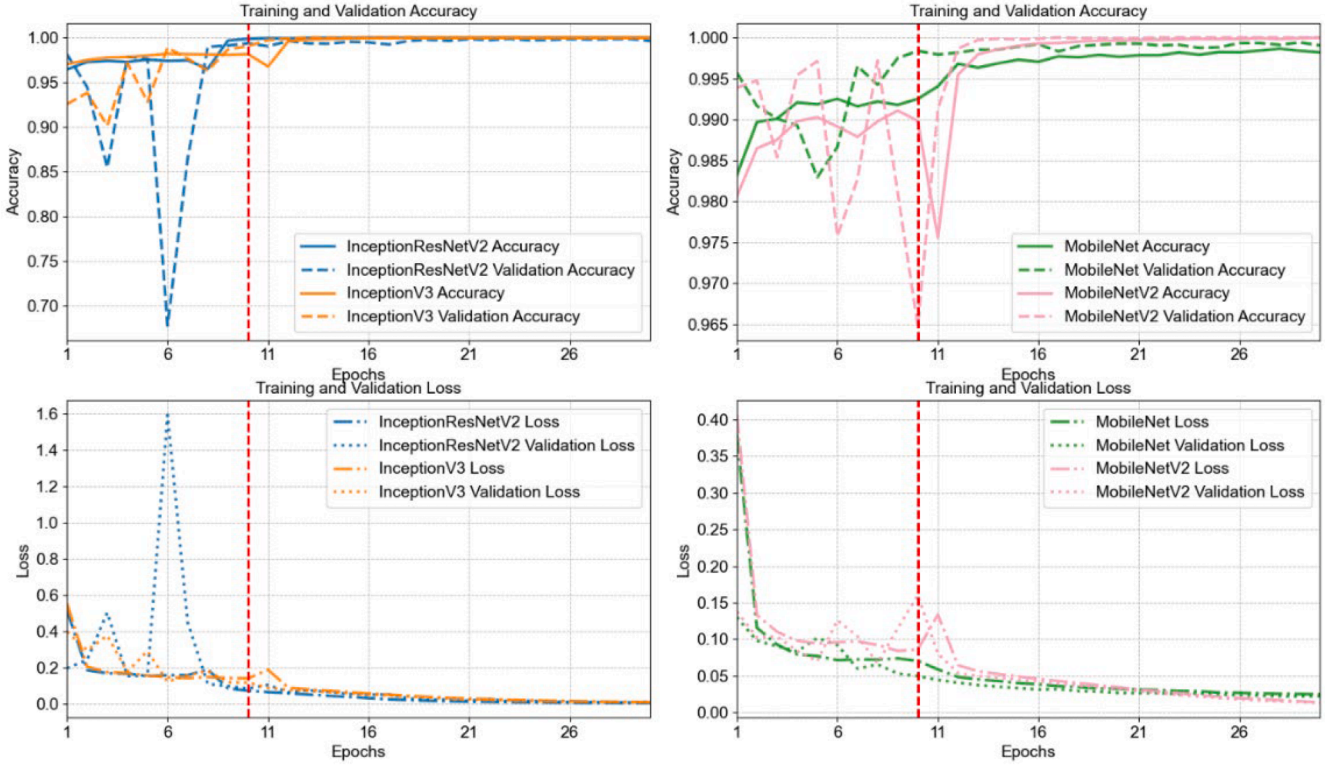


Fig. 11. Visualization of the accuracy and loss of the models applying the four test set, MobileNetV2 Model, MobileNet Model, InceptionV3 Model, Inception-ResNetV2 Model.

Table 8
Results of the architectures in Test 5.

Models	Training Set				Test Set				Val Loss	Epochs
	A	P1	R	F1Sc	A	P1	R	F1Sc		
InceptionV4	0.17	0.17	0.17	0.17	0.17	0.17	0.17	0.17	0.058	7
AlexNet	0.16	0.16	0.16	0.16	0.16	0.16	0.16	0.16	0.013	21
Linet5	0.17	0.03	0.17	0.05	0.17	0.03	0.17	0.05	0.013	21

Table 9
Results of the architectures in Test 6.

Models	Training Set				Test Set				Val Loss	Epochs
	A	P1	R	F1Sc	A	P1	R	F1Sc		
InceptionV4	0.19	0.19	0.19	0.19	0.19	0.19	0.19	0.19	0.012	17
AlexNet	0.17	0.17	0.17	0.17	0.17	0.17	0.17	0.17	0.034	20
Linet5	0.17	0.03	0.17	0.05	0.17	0.03	0.17	0.05	1.795	6

reduce training time (by up to 95 %, with epochs between 11 and 19 compared to the fixed 200 in Test 1) in models such as MobileNet and InceptionV3, while maintaining outstanding metrics (F1-Score $\geq$ 0.98) and improving generalization by reducing validation losses by up to 100 times (MobileNet: Val Loss 0.000084 vs. 0.001179; DenseNet121: Val MSE 0.000016 vs. 0.001915). However, their use also accentuated weaknesses in unstable architectures: MobileNetV2 and DenseNet121 experienced significant drops in F1-Score (from 1.0 to 0.86 and from 0.97 to 0.90, respectively), while VGG16 collapsed from 0.97 to 0.34, demonstrating that early stopping can prematurely halt learning if an early improvement threshold is not met.Thus, while these techniques notably optimize certain models (InceptionResNetV2 improved from 0.97 to 0.98), their efficacy critically depends on baseline architectural robustness, as they do not compensate for design flaws in variants such as MobileNetV3 or ResNet, where improvements were minimal (F1-Score from 0.45 to 0.49) or negligible. This duality highlights that, despite their utility in preventing overfitting and accelerating convergence, their application requires rigorous validation of learning curves and strategic architecture selection to mitigate underfitting risks.

In Test 3, the Batch Normalization technique is used, which helps stabilize and speed up the training of neural networks, and the L2 Regularization technique, which helps prevent overfitting by penalizing large weights. In addition, they are trained with other techniques, such as the Early Stopping technique, the Plateau Learning Rate Reduction technique, the Transfer Learning technique, and four layers were added to the model, as shown in Fig. 7. In addition, the Data Augmentation technique was used, training each model for 200 epochs.

In Table 6, the results of test three can be seen, The implementation of Batch Normalization (BN) and L2 Regularization in Test 3 yielded significant advantages in terms of efficiency and generalization for

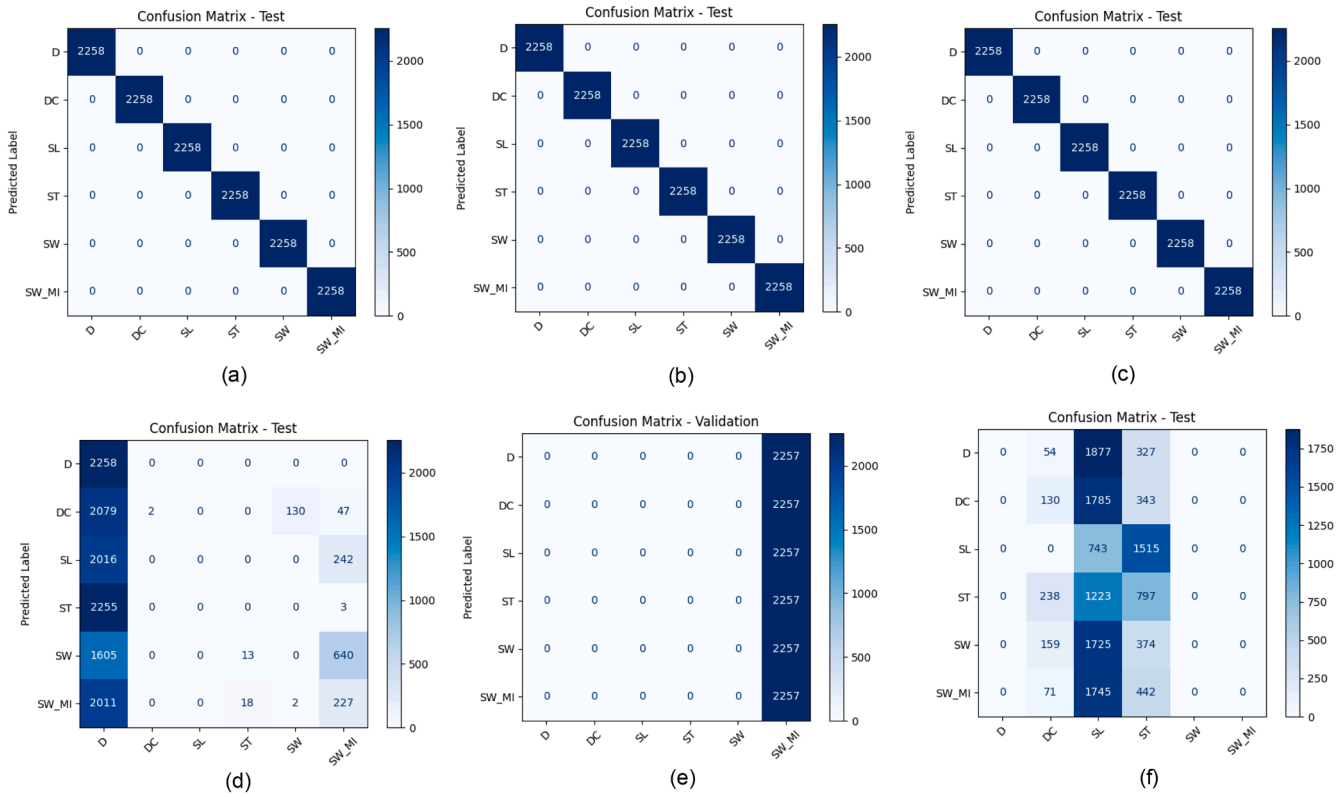


Fig. 12. Confusion matrices of the three best models with accuracy of 1.0, and confusion matrices of the three models that obtained accuracy below 0.2 (a) Movilnet, (b) InceptionV3, (c) InceptionResNetV2, (d) ResNet50, (e) ResNet152V2, (f) AlexNet.

specific architectures, while also presenting critical drawbacks in more complex models. Among the most notable benefits, the inclusion of BN substantially accelerated convergence, reducing the required training epochs by up to 82 % in models such as MobileNetV3Small (from 44 to 8 epochs) and NasNetLarge (from 16 to 8 epochs). Concurrently, the combination of BN and L2 enhanced generalization in lightweight architectures: MobileNetV3Large exhibited an 11 % increase in accuracy (0.49 to 0.60) and a 9 % improvement in F1-Score (0.49 to 0.58), while VGG16 achieved a 12 % gain in F1-Score (0.34 to 0.46). These results demonstrate improved overfitting control. However, critical disadvantages related to over-regularization and metric degradation were identified. Complex models such as InceptionV3 and InceptionResNetV2 experienced increases of up to 60-fold in validation loss (val_loss) (InceptionV3: 0.0014 to 0.062) and 4-fold in validation Mean Squared Error (val_MSE) (InceptionResNetV2: 0.0002 to 0.0008), indicating optimization instability. More concerningly, the ResNet family exhibited alarming regressions: all variants maintained marginal accuracy (~0.17) but showed increased Recognition Error Rate (val_ER), as exemplified by ResNet50V2 (0.9342 to 0.9724). This reveals an incompatibility with the implemented techniques.

In Test 4, the fine-tuning technique is applied in combination with the methods used in the third test set. The code implements an efficient fine-tuning strategy by integrating transfer learning with adaptation to a new dataset. By freezing the initial layers of the model and subsequently unfreezing from layer 100 for fine-tuning. Table 7 shows the validation results of the models in test set four. The implementation of a selective fine-tuning strategy in Test 4 presents significant advantages but also critical disadvantages, highlighting the complex and architecture-dependent nature of this technique.

The primary advantage lies in the substantial improvement or maintenance of high performance in specific models. MobileNetV2 exemplifies this, demonstrating an extraordinary improvement, with its F1-Score increasing from 0.88 in Test 3 to an excellent 0.99 in Test 4, accompanied by a significant reduction in validation loss (Val Loss: 0.026 to 0.011). NasNetLarge also showed notable improvement, increasing its F1-Score from 0.92 to 0.97 with a lower Val Loss (0.047 to 0.008). Similarly, the MobileNetV3 variants (Small and Large) achieved much more stable and efficient convergence, drastically reducing their Val Loss (from 21.583 to 0.109 and from 5.464 to 0.098, respectively), although their precision and recall metrics remained moderate. Already robust models like MobileNet, InceptionV3, and InceptionResNetV2 maintained near-perfect performance (Accuracy (A), Precision (P), Recall (R), F1-Score ≈ 1.0) achieved in Test 3, often with a slightly lower Val Loss. This corroborates that fine-tuning can further optimize well-adapted models without compromising their accuracy. However, the technique revealed pronounced disadvantages in other architectures, particularly within the DenseNet family. Models DenseNet201, DenseNet169, and DenseNet121 suffered catastrophic performance collapse. For instance, DenseNet201 dropped from an F1-Score of 0.96 in Test 3 to merely 0.15 in Test 4, while also exhibiting very high Relative Error (Val ER: 0.8045) and Mean Squared Error (Val MSE: 0.8019) despite a low Val Loss (0.003). This indicates severe overfitting or convergence to a non-optimal local minimum during fine-tuning, potentially exacerbated by their dense connectivity. VGG16 also experienced degradation, with its F1-Score decreasing from 0.46 to 0.32. In contrast, ResNet architectures (50, 101V2, 152V2, 50V2) showed no improvement in predictive capability (Accuracy stagnant at 0.17, F1-Score at 0.05), despite significant reductions in Val Loss (e.g., ResNet50: 7.107 to 0.246). This suggests that fine-tuning improved convergence but failed to resolve their underlying inability to generalize for this task. Additionally, Test 4 incurred higher computational costs, with a notable increase in the number of training epochs required for most models (e.g., MobileNet: 11 to 26 epochs, InceptionV3: 12 to 27 epochs, ResNet101V2: 18 to 43 epochs).

Fig. 10 shows a radar chart comparing the ResNet50V2 and VGG16 networks across the four training tests proposed in the first approach. It

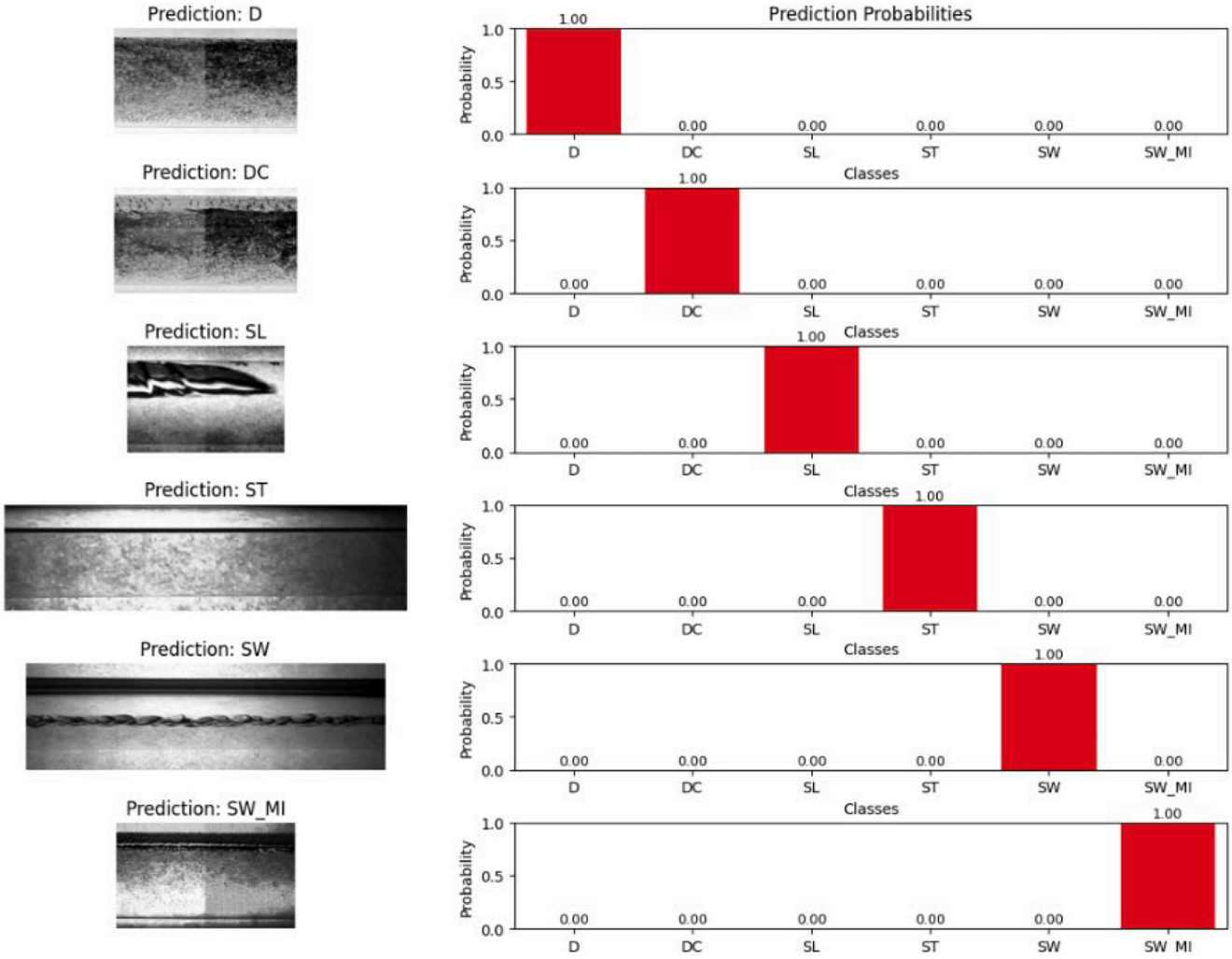


Fig. 13. Bar chart of MobileNet model predictions of Test 4.

can be observed that in Test 1, they achieved an accuracy of 0.97 with 200 epochs. When applying the Early Stopping technique and the Plateau Learning Rate Reduction technique, the performance drops to 0.14 in the case of the ResNet50V2 network and to 0.42 in the case of the VGG16 network. This is due to the premature activation of Early Detention. If the model has not yet reached its optimal performance, Early Detention could interrupt training before it achieves adequate learning, or it also usually happens if there is an abrupt change in the learning dynamics.

The analysis of the specificity parameter, which measures a model's ability to correctly identify negative cases and avoid false positives, reveals fundamental aspects of the discriminative robustness of both models. The results shown in Fig. 10 demonstrate that VGG16 maintains higher and more stable specificity compared to ResNet50V2 across the four experimental tests, suggesting an inherent capacity to minimize misclassifications even under adverse training conditions. The impact of regularization techniques was markedly different; the deeper and more complex ResNet50V2 architecture experienced a more severe degradation in this metric due to its greater convergence time requirements, whereas the simpler sequential architecture of VGG16 allowed it to better adapt to anticipated interruptions, maintaining consistent specificity values.

The MobileNet, InceptionV3, and InceptionResNetV2 models achieved an accuracy of 1.0 in Test 4, standing out in performance among the six evaluated tests without showing signs of overfitting. In the case of MobileNetV2, the accuracy was 0.9. Fig. 11 displays the accuracy and loss graphs of these four networks, applying the fine-tuning technique. The red line in Fig. 11 marks the start of fine-tuning: after an initial training phase with the lower layers (up to layer 100) frozen to preserve the patterns extracted from ImageNet, the upper layers (from layer 100 onward) are unfrozen to retrain them with the biphasic flow database. This transition allows the model to adjust to specific features of the new domain, reflected in the graphs as a sustained increase in accuracy and a reduction in loss during both training and validation. The stability of the metrics after the red line, combined with rapid convergence in few epochs, confirms that the process optimizes performance without overfitting, maintaining low computational cost due to the controlled approach of selectively unfreezing layers.

The results of tests 5 and 6 can be visualized in Tables 8 and 9, respectively. When applying these models to the dense-gas/liquid two-phase flow data set, no good results are obtained for the A, P1, R, and F1 Sc metrics.

In Fig. 12 one contrasts the neural networks that achieved the best evaluation metrics with those that showed the worst evaluation metrics among the six tests analyzed. In the high-performance group, a) MobileNet, b) InceptionV3, and c) InceptionResNetV2 stand out, achieving an accuracy of 1.0 on the test data, while the low-performance group includes d) ResNet50, e) ResNet152V2, and f) AlexNet, with an accuracy below 0.2. The confusion matrices displayed correspond exclusively to the test data, where the rows represent the actual flow regimes and the columns the predicted ones. Each numerical cell reflects the number of classified images: diagonal values indicate correct predictions (a match

between the actual and predicted class), and off-diagonal values indicate errors.

For the models with the best metrics (a, b, c), the matrices exhibit a perfect diagonal with no errors (all off-diagonal elements are zero), confirming that they correctly classified all images in the test set. This result suggests that MobileNet, InceptionV3, and InceptionResNetV2 have exceptional predictive capability for gas-dense/liquid two-phase flow classification.

In contrast, the models with the worst metrics (d, e, f) display confusion matrices with a massive dispersion of prediction errors outside the diagonal. For example, ResNet50, ResNet152V2, and AlexNet demonstrated a clear inability to distinguish between flow regimes, assigning most images to incorrect classes. This behavior suggests underfitting, where the models failed to learn relevant patterns from the data, either due to insufficient architectural capacity, limited training datasets, or poorly configured hyperparameters (such as inadequate learning rates or insufficient training epochs).

The ResNet50, ResNet152V2, MobileNetV3Small, MobileNetV3Large, ResNet101V2, InceptionV4, AlexNet, and Linet5 models did not exceed an accuracy of 0.2, achieving the lowest results in training for dense-gas/liquid two-phase flow pattern recognition.

In Fig. 13, one shows the flow pattern predictions by selecting some images from the test data set for the MobileNet model, which shows the highest performance and lowest computational cost. Also shown is the prediction probability for each flow pattern, which is the result of the SoftMax layer. The red histogram indicates the correct prediction probability.

4. Conclusions

The present study has comprehensively addressed the identification of dense-gas/liquid two-phase flow patterns in horizontal pipes using a combination of experimentation and advanced artificial intelligence techniques. The conclusions derived from the findings are detailed below.

The study presents a novel model by applying modified convolutional neural networks (CNN), integrating transfer learning with techniques such as data augmentation, early stopping, batch normalization, L_2 regularization, and fine-tuning, to identify gas-dense/liquid flow patterns. The methodology was evaluated across six experimental tests, where architectures like MobileNet, InceptionV3, and InceptionResNetV2 demonstrated optimal accuracy (1.0) in Test 4. This performance was achieved by adapting pretrained layers to domain-specific features, minimizing overfitting, and reducing computational costs from 200 to a maximum of 11–27 epochs. In contrast, models such as ResNet50, ResNet152V2, and AlexNet exhibited poor performance (accuracy $<$ 0.2), linked to underfitting and inadequate architectural complexity, as evidenced by confusion matrices with significant dispersion.

The implementation of early stopping and dynamic learning rate adjustment optimized efficiency in certain architectures; for instance, the MobileNet network reduced its training from 17 to 11 epochs. However, it degraded performance in others, such as ResNet50V2 and VGG16 networks, due to premature training interruptions. Batch normalization and L_2 regularization stabilized the learning process, though their effectiveness varied depending on the network structure.

The findings of this study highlight three fundamental pillars for the effective identification of two-phase flow patterns. First, domain-specific adaptation through fine-tuning allows for the transfer of generic patterns learned from extensive databases (e.g., ImageNet) and refines them for the target domain, maximizing accuracy without compromising the model's generalization. Second, architectural selection emerges as a critical factor: lightweight and modular topologies, such as MobileNet, consistently outperformed deep and static networks (e.g., ResNet), due to their intrinsic computational efficiency and flexibility in adapting to the dynamic and heterogeneous characteristics of gas-liquid flow. Finally, the need for a balance between regularization techniques and data complexity is evident, as an unbalanced use of methods such as early stopping or L_2 penalty can inhibit the model's learning capacity or suppress critical features, degrading its predictive performance.

CRediT authorship contribution statement

July A. Gómez-Camperos: Writing – review & editing, Writing – original draft, Visualization, Formal analysis, Data curation. **Carlos M. R. Diaz:** Supervision, Investigation, Conceptualization. **Marlon M. Hernandez-Cely:** Validation, Supervision, Methodology, Conceptualization. **Oscar M. H. Rodriguez:** Supervision, Formal analysis, Conceptualization. **Aldo Pardo-Garcia:** Supervision, Data curation, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

Oscar Rodriguez would like to thank CNPq for the productivity grant (proc. 311057/2020-9). Carlos Ruiz gratefully acknowledges the support of the Ministry of Science, Technology, and Innovation of Colombia.

Appendix

Table.

Table
Video cropping process pseudocode.

Step	Category	Pseudocode Statement	Description
1	Initialization	IMPORT cv2	Import OpenCV library
2	Configuration	SET input_path	Define input video path
		SET output_path	Define output video path
3	Input Handling	input_video	Initialize video capture
4	Validation	IF input_video NOT OPEN: PRINT "Error: Cannot open input video"; EXIT	Verify successful video access
5	Metadata Extraction	fps = GET_FPS(input_video)	Retrieve frames per second
		width = GET_WIDTH(input_video)	Get original frame width
		height = GET_HEIGHT(input_video)	Get original frame height
6	Crop Configuration	SET top_crop = 600	Pixels to remove from top

(*continued on next page*)

Table (*continued*)

Step	Category	Pseudocode Statement	Description
		SET bottom_crop = 550	Pixels to remove from bottom
		new_height = height - top_crop - bottom_crop	Calculate new frame height
7	Output Setup	codec = GET_CODEC('mp4v')	Specify video compression format
		output_video = CREATE_VIDEO(output_path, codec, fps, (width, new_height))	Initialize output writer
8	Validation	IF output_video NOT OPEN: PRINT "ERROR"; EXIT	Verify output file creation
9	Frame Processing	WHILE TRUE:Start frame processing loop	Start frame processing loop
		├─ status, frame = READ_FRAME(input_video) Read next frame	Read next frame
		├─ IF NOT status: BREAK	Exit loop at end of video
10	Transformation	├─ cropped_frame = frame[top_crop: height - bottom_crop,:]	Crop top/bottom regions
11	Output	└─ WRITE_FRAME(output_video, cropped_frame)	Write processed frame to output
12	Cleanup	RELEASE(input_video)	Release input resources
		RELEASE(output_video)	Release output resources
13	Completion	PRINT "Cropped video saved successfully"	Confirm successful execution

Data availability

No data was used for the research described in the article.

References

AL-Dogail, A.S., Gajbhiye, R.N., 2021. Effects of density, viscosity and surface tension on flow regimes and pressure drop of two-phase flow in horizontal pipes. J. Pet. Sci. Eng. 205 (March), 108719. https://doi.org/10.1016/j.petrol.2021.108719.

AL-Dogail, A.S., Gajbhiye, R.N., 2021. Effects of density, viscosity and surface tension on flow regimes and pressure drop of two-phase flow in horizontal pipes. J. Pet. Sci. Eng. 205 (April), 108719. https://doi.org/10.1016/j.petrol.2021.108719.

Arabi, A., Salhi, Y., Zenati, Y., Si-Ahmed, E.K., Legrand, J., 2020. On gas-liquid intermittent flow in a horizontal pipe: influence of sub-regime on slug frequency. Chem. Eng. Sci. 211, 115251. https://doi.org/10.1016/j.ces.2019.115251.

Belt, R., Smith, I.E., Staff, G., Lawrence, C., 2024. Growth of solitary slugs in long pipes. Int. J. Multiph. Flow 178 (May), 104883. https://doi.org/10.1016/j.ijmultiphaseflow.2024.104883.

Carneiro, J.N.E., et al., Oct. 2015. Numerical simulations of high CO_2 content flows in production wells, flowlines and risers. In: Proceedings of the OTC Brasil, OTC, pp. 1522–1533. https://doi.org/10.4043/26231-MS.

das Neves, D.A., Vieira, S.C., Cenzi, J.R., de Paula, A.S., de Castro, M.S., Fabro, A.T., 2024. Identification of the flow pattern from the experimental pressure signal in horizontal pipes carrying two-phase flows. Exp. Therm. Fluid Sci. 154 (March 2023), 111141. https://doi.org/10.1016/j.expthermflusci.2024.111141.

Dong, F., Zhang, S., Shi, X., Wu, H., Tan, C., 2021. Flow regimes identification-based multidomain features for gas-liquid two-phase flow in horizontal pipe. IEEE Trans. Instrum. Meas. 70, 1–11. https://doi.org/10.1109/TIM.2021.3089767.

Gonçalves, J.L., Mazza, R.A., 2024. Slug flow parameters applied to dynamic riser analysis. Int. J. Multiph. Flow 171 (November 2023). https://doi.org/10.1016/j.ijmultiphaseflow.2023.104670.

He, K., Zhang, X., Ren, S., Sun, J., 2016. Identity mappings in deep residual networks. Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics) 9908, 630–645. https://doi.org/10.1007/978-3-319-46493-0_38. LNCS.

Howard A.G., et al., "MobileNets: efficient convolutional neural networks for mobile vision applications," 2017.

Huang, G., Liu, Z., Van Der Maaten, L., Weinberger, K.Q., 2017. Densely connected convolutional networks. In: Proceedings of the 30th IEEE Conference on Computer Vision Pattern Recognition, CVPR 2017, 2017-Janua, pp. 2261–2269. https://doi.org/10.1109/CVPR.2017.243.

Khan, U., Pao, W., Pilario, K.E., Sallih, N., 2024. Flow regime classification using various dimensionality reduction methods and AutoML. Eng. Anal. Bound. Elem. 163 (November 2023), 161–174. https://doi.org/10.1016/j.enganabound.2024.03.006.

Kuang, B., Nnabuife, S.G., Whidborne, J.F., Sun, S., Zhao, J., Jenkins, K., 2024. Self-supervised learning-based two-phase flow regime identification using ultrasonic sensors in an S-shape riser. Expert. Syst. Appl. 236 (August 2023), 121414. https://doi.org/10.1016/j.eswa.2023.121414.

Liu, W., Tan, C., Dong, F., 2021. Doppler spectrum analysis and flow pattern identification of oil-water two-phase flow using dual-modality sensor. Flow Meas. Instrum. 77 (March 2020), 101861. https://doi.org/10.1016/j.flowmeasinst.2020.101861.

Luo, C., et al., 2023. Experimental and modeling investigation on gas-liquid two-phase flow in horizontal gas wells. J. Energy Resour. Technol. Trans. ASME 145 (1), 1–13. https://doi.org/10.1115/1.4055223.

Miseta, T., Fodor, A., Vathy-Fogarassy, Á., 2024. Surpassing early stopping: a novel correlation-based stopping criterion for neural networks. Neurocomputing 567 (November 2023), 127028. https://doi.org/10.1016/j.neucom.2023.127028.

Mohammed, S., Abdulkareem, L., Roshani, G.H., Eftekhari-Zadeh, E., Haso, E., 2022. Enhanced multiphase flow measurement using dual non-intrusive techniques and ANN model for void fraction determination. Processes 10 (11). https://doi.org/10.3390/pr10112371.

Mohmmed, A.O., Al-Kayiem, H.H., Osman, A.B., 2021. Investigations on the slug two-phase flow in horizontal pipes: past, presents, and future directives. Chem. Eng. Sci. 238, 116611. https://doi.org/10.1016/j.ces.2021.116611.

Mumuni.A, and F. Mumuni, "Data augmentation with automated machine learning: approaches and performance comparison with classical data augmentation methods," *arXiv.org, abs/2403.08352*, pp. 1–28, 2024.

Nakamura, K., Derbel, B., Won, K.J., Hong, B.W., 2021. Learning-rate annealing methods for deep neural networks. Electron 10 (16), 1–12. https://doi.org/10.3390/electronics10162029.

Nie, F., Wang, H., Song, Q., Zhao, Y., Shen, J., Gong, M., 2022. Image identification for two-phase flow patterns based on CNN algorithms. Int. J. Multiph. Flow 152 (February), 104067. https://doi.org/10.1016/j.ijmultiphaseflow.2022.104067.

Nnabuife, S.G., Kuang, B., Whidborne, J.F., Rana, Z.A., 2023. Development of gas-liquid flow regimes identification using a noninvasive ultrasonic sensor, belt-shape features, and convolutional neural network in an S-shaped riser. IEEe Trans. Cybern. 53 (1), 3–17. https://doi.org/10.1109/TCYB.2021.3084860.

Onuoha, M.D.U., Duan, M., Xu, X., Liu, J., 2022. Numerical modelling of the interaction between severe slugging and dynamic response of deepwater risers: a time-domain ANHE-FD based solution. Int. J. Non Linear Mech. 138 (July 2021), 103834. https://doi.org/10.1016/j.ijnonlinmec.2021.103834.

Qiao, W., Guo, H., Huang, E., Chen, H., Lian, C., 2023. Two-phase flow pattern identification by embedding double attention mechanisms into a convolutional neural network. J. Mar. Sci. Eng. 11 (4). https://doi.org/10.3390/jmse11040793.

Qiumei, Z., Yukun, H., Fenghua, W., Zhang, P., Chao, L., 2023. Liqnet: a real-time monitoring network for two-phase flow patterns. Flow Meas. Instrum. 90 (December 2022), 102313. https://doi.org/10.1016/j.flowmeasinst.2023.102313.

Quintino, A.M., da Fonseca Junior, R., Rodriguez, O.M.H., 2023. Experimental study of liquid/dense-gas pipe flow. Geoenergy Sci. Eng. 230 (March). https://doi.org/10.1016/j.geoen.2023.212179.

Salgado, C.M., de F. Dam, R.S., de C. Conti, C., Salgado, W.L., 2021. Three-phase flow meters based on X-rays and artificial neural network to measure the flow compositions. Flow Meas. Instrum. 82 (October). https://doi.org/10.1016/j.flowmeasinst.2021.102075.

Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., Chen, L.C., 2017. MobileNetV2: inverted residuals and linear bottlenecks. Proc. IEEE Conf. Comput. Vis. Pattern Recognit. 4510–4520. https://doi.org/10.1109/CVPR.2018.00474.

Shi, X., Tan, C., Dong, F., Santos, E.N.D., Da Silva, M.J., 2021. Conductance sensors for multiphase flow measurement: a review. IEEE Sens. J. 21 (11), 12913–12925. https://doi.org/10.1109/JSEN.2020.3042206.

Shin, H.C., Kim, S.H., Shah, Y., Kim, S.M., 2024. An experimental study on air-oil flow patterns in horizontal pipes using two synthetic oils. Int. J. Heat. Mass Transf. 226 (March), 125459. https://doi.org/10.1016/j.ijheatmasstransfer.2024.125459.

Simonyan, K., Zisserman, A., 2015. Very deep convolutional networks for large-scale image recognition. In: Proceedings of the 3rd International Conference on Learning Representations ICLR 2015 - Conference Track, pp. 1–14.

Tandoh, H., Nnabuife, S.G., Cao, Y., Lao, L., Whidborne, J., 2022. Slug flow regime in a flowline with U-shape riser. Case Stud. Chem. Environ. Eng. 6 (August), 100265. https://doi.org/10.1016/j.cscee.2022.100265.

Vaz, R.G., von Hohendorff Filho, J.C., Botechia, V.E., Schiozer, D.J., October 2021. Modeling subsea gas-liquid separation using proxy for phase separation for pre-salt reservoirs with high GOR. J. Pet. Sci. Eng. 212, 2022. https://doi.org/10.1016/j.petrol.2022.110267.

Wang, Y., Yu, Y., Liu, Z., Chang, Y., Zhao, X., Wang, Q., 2024. Experimental study on gas-liquid two-phase stratified flow at high pressure in a horizontal pipe. Energies (Basel) 17 (5). https://doi.org/10.3390/en17051056.

Xu, H., Tang, T., Zhang, B., Liu, Y., 2022. Identification of two-phase flow regime in the energy industry based on modified convolutional neural network. Prog. Nucl. Energy 147 (April 2021), 104191. https://doi.org/10.1016/j.pnucene.2022.104191.

Yan, Q., Li, D., Wang, K., Zheng, G., 2024. Study on the hydrodynamic evolution mechanism and drift flow patterns of pipeline gas–liquid flow. Processes 12 (4). https://doi.org/10.3390/pr12040695.

Yang, G., et al., 2024. Numerical analysis on two-phase flow-induced vibrations at different flow regimes in a spiral tube. Nucl. Eng. Technol. 56 (5), 1712–1724. https://doi.org/10.1016/j.net.2023.12.025.

Yaqub, M.W., Pendyala, R., 2024. Flow regime transition maps and pressure loss prediction of gas, oil and water three-phase flow in the vertical riser downstream 90° bend using data driven approach. Digit. Chem. Eng. 12 (April), 100174. https://doi.org/10.1016/j.dche.2024.100174.

Yuan, X., Feng, Z., Norton, M., Li, X., 2019. Generalized batch normalization: towards accelerating deep neural networks. In: Proceedings of the 33rd AAAI Conference on Artificial Intelligence AAAI 2019, 31st Innovative Applications of Artificial Intelligence Conference IAAI 2019 9th AAAI Symposium on Educational Advances in Artificial Intelligence EAAI 2019, pp. 1682–1689. https://doi.org/10.1609/aaai.v33i01.33011682.

Zhang, D., Ouyang, B., Luo, Z.H., 2024. Identification of gas-solid flow regimes using convolutional neural network techniques. Powder Technol. 442 (March), 119848. https://doi.org/10.1016/j.powtec.2024.119848.

Zhang, J., Wei, X., Wang, Z., 2023. The recognition algorithm of two-phase flow patterns based on GoogLeNet+5 coord attention. Micromachines (Basel) 14 (2). https://doi.org/10.3390/mi14020462.

Zhao, X., Wang, L., Zhang, Y., Han, X., Deveci, M., Parmar, M., 2024. A Review of Convolutional Neural Networks in Computer Vision, 57. Springer, Netherlands. https://doi.org/10.1007/s10462-024-10721-6.

Zoph, B., Vasudevan, V., Shlens, J., Le, Q.V., 2018. Learning transferable architectures for scalable image recognition. Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern. Recognit. 8697–8710. https://doi.org/10.1109/CVPR.2018.00907.

Zou, S., Guo, L., Yao, T., 2021. Upstream-flow-based mechanisms for global flow regime transition of gas/liquid two-phase flow in pipeline-riser systems. Chem. Eng. Sci. 240, 116542. https://doi.org/10.1016/j.ces.2021.116542.