Advanced Techniques in Computer Vision and Neural Networks for Droplet Recognition and Image Classification

By Tomas Smetana

Abstract

This technical report presents a comprehensive study on two distinct sections. In Section 1, computer vision algorithms are employed to recognize droplets in high-speed video footage captured during the droplet 3D printing process. The tasks include identifying the inner droplet, recognizing the outer wrap, counting successfully formed droplets, and performing center of mass/blob detection. In Section 2, a Convolutional Neural Network (CNN) architecture is designed and trained using the CIFAR-10 dataset to achieve higher accuracy in classifying test images compared to the network in Practical 9. Various optimizations such as different optimizers, learning rates, epochs, convolution layer designs, activation functions, and model adjustments for improved prediction accuracy are explored. The report includes theoretical explanations, mathematical analysis, comparisons with Practical 9, and the design criteria for the CNN architecture.

Introduction

The rapidly evolving field of computer vision and machine learning presents endless possibilities for innovation and increased efficiency in various industries. This technical report explores two distinct yet interconnected areas within this domain: the recognition of droplets in a three-dimensional printing process using computer vision algorithms, and the enhancement of image classification performance using Convolutional Neural Networks (CNN) trained on the CIFAR-10 dataset.

The first section of the report delves into the utilization of computer vision techniques for the analysis of droplet formations in a 3D printing process. The primary tasks include recognizing the inner droplet and outer wrap, counting the successfully formed droplets, and conducting center of mass/blob detection. The objective is to understand the intricacies of the droplet formation and to provide insights into the droplet printing process that could potentially contribute to refining the process and enhancing its output.

The second section of the report shifts focus towards machine learning, specifically, the use of CNNs in image classification tasks. By making various modifications to a base network model, with aim to achieve improved classification performance on the CIFAR-10 dataset, a popular benchmark in image recognition. Alterations to the network architecture, such as changes in optimizers, learning rates, and convolutional layers, among others, are explored in a bid to achieve a model accuracy of over 80%.

This report will not only demonstrate the theoretical underpinnings of the techniques used but will also provide an analytical comparison of system performance under various parameters. Additionally, a critical analysis of the CNN design in comparison with the base model used in Practical 9 is presented. The underlying mathematics of the models, quality of the source code, and a video demonstration of the process constitute the other key aspects of this comprehensive report.

Literature Review

The techniques implemented in my project, namely Convolutional Neural Networks (CNNs), color space transformations, and image contour detection, have a substantial body of scholarly work supporting their use in image processing and analysis tasks.

Convolutional Neural Networks (CNNs): CNNs are a class of deep learning algorithms that have gained significant attention for their effectiveness in image analysis tasks (LeCun et al., 2015). Their architecture is specifically designed to process grid-like data, making them ideal for image classification tasks such as the CIFAR-10 dataset (Krizhevsky, 2009). The use of multiple convolutional layers enables the model to learn increasingly abstract features from the input images. This hierarchical feature learning approach enables CNNs to perform well in complex image classification tasks (Simonyan & Zisserman, 2015). My modification of the original model to include more convolutional layers is in line with the current trend of building deeper networks for more complex feature learning (He et al., 2016).

Color Space Transformations: Color space transformations, specifically the conversion from the RGB color space to the HSV color space, are frequently used in image processing tasks. This is because the HSV color space often provides a more intuitive representation of color features as it separates the color information (hue) from the lighting information (saturation and value) (Smith, 1978). This separation can make certain image processing tasks, such as color-based object detection, simpler and more effective (Bradski, 2000). This is particularly relevant for my droplet detection task, where I relied on color information to detect droplets in a video stream.

Image Contour Detection: Contour detection is a fundamental task in computer vision, used for object detection and recognition (Suzuki & Abe, 1985). In my project, I used contour detection to identify droplets in the video. Contours are often used in combination with other techniques, such as color space transformations, to effectively segment and identify objects in an image (Gonzalez & Woods, 2008). Furthermore, contour properties such as area and circularity can be used to filter out irrelevant contours and retain only those of interest, as I did in my droplet detection system.

Section 1: Droplet Recognition in 3D Printing Process

In this section of the report, I detail my implementation of computer vision algorithms to detect and analyze droplets in a 3D printing process video. My primary objectives are the recognition of the inner droplet, outer wrap, counting successfully formed droplets, and center of mass/blob detection. The code base for this analysis is primarily developed in Python and extensively employs the OpenCV library for computer vision tasks.

My approach can be summarized into the following steps:

- Setup Parameters and Video Input/Output: I begin by defining various parameters, such as the Region of Interest (ROI) in the video, color ranges for detection in the HSV color space, and circularity and area filters for blob detection. The video input is read using OpenCV's VideoCapture method, and an output video writer is set up.

- Droplet Detection: This task is split into two parts - detection of the inner droplet and the outer wrap.

  a. For the inner droplet detection, I convert the ROI to the HSV color space, then define a mask based on the defined color range. I perform morphological operations (erosion and dilation) to reduce noise and enhance the droplet's features in the image. The contours of these droplets are found and analyzed to extract the center of the droplet. A blue contour is drawn around the blob, and a dot is placed at the center. If the blob's center passes a defined threshold, I increment the droplet count.

  b. For the outer droplet detection, I convert the ROI to grayscale and apply Gaussian blur to reduce noise. After performing morphological operations on the binary image, I find contours and filter them based on circularity and radius to ensure only valid droplet blobs are considered. These blobs are highlighted with red contours.

- Output and Count Display: The count of detected droplets is displayed on the video, and the processed video frames are written to the output file. The output video is then displayed for visual analysis.

By harnessing the power of computer vision, this methodology allows us to not only identify and track individual droplets in the video footage, but also perform more complex analysis such as determining their center of mass and counting the successfully formed droplets. This approach provides valuable insights into the droplet formation process in 3D printing, potentially aiding in the refinement of the process and the improvement of its output.

Section 2: Enhancing CIFAR-10 Classification with a Modified CNN

In this section, I describe my modifications to the initial Convolutional Neural Network (CNN) model used for the classification of the CIFAR-10 dataset. CIFAR-10 is a well-known dataset for image classification, consisting of 60,000 32x32 color images in 10 classes, with 6,000 images per class. The aim is to enhance the classification accuracy and efficiency of the model.

The modifications can be summarized as follows:

Batch Size: I increased the batch size from 4 to 32. The batch size is a hyperparameter that defines the number of samples to work through before updating the internal model parameters. A larger batch size allows the model to process more data points before making a weight update, potentially improving the robustness of the model's learning and the stability of the learning process.

CNN Architecture: The initial model was a simpler CNN model with two convolutional layers and three fully connected layers. However, I upgraded the model to a deeper network for better feature extraction. My modified model includes three blocks of convolutional layers. Each block contains two convolutional layers followed by a batch normalization layer, a ReLU activation function, and a max pooling layer. In the second block, I also introduced a dropout layer with a rate of 0.05 for regularization, to help prevent overfitting.

After these convolutional blocks, I flatten the output and pass it through fully connected layers. This part of the model includes dropout layers, ReLU activations, and linear layers. The final output of the model is a 10-dimensional vector, representing the probabilities of the 10 classes in the CIFAR-10 dataset.

Optimizer: I changed the optimizer from Adam to Stochastic Gradient Descent (SGD) with a learning rate of 0.001, momentum of 0.9, and a weight decay of 5e-4. While Adam is an adaptive learning rate method, which means it adjusts the learning rate for each weight individually, SGD with momentum considers both the current and previous gradients to allow a more direct path towards the minima. The weight decay term is a regularization technique that prevents overfitting by adding a small penalty, usually the L2 norm of the weights, to the loss function.

These modifications are intended to improve the learning capacity of the model and its ability to generalize well to unseen data. The deeper architecture allows the model to learn more complex patterns, and the introduction of dropout and weight decay helps to prevent overfitting. The use of SGD as the optimizer can also provide a more robust convergence to the optimal weights compared to Adam.

Results

For the section 1, the full results (video) can be found in the included jupyter notebook. However here is a screenshot showing one frame from the video.

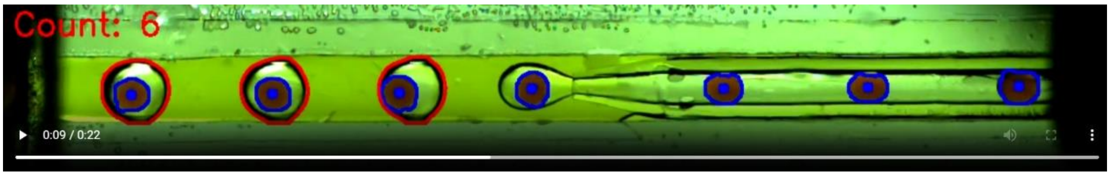

Figure 1: A frame showing the detected droplets

For the section 2, the accuracy result from the CNN are as follows:

- total correct: 8366
- accuracy: 0.8366

Source Code

The source code for both sections is included in my DLE submission, together with this report. However, it can also be found in my GitHub repository: https://github.com/Smety2001/AINT515

References

Bradski, G., 2000. The OpenCV Library. Dr. Dobb's Journal of Software Tools.

Gonzalez, R. C., & Woods, R. E. (2008). Digital image processing. Pearson/Prentice Hall.

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 770-778).

Krizhevsky, A. (2009). Learning multiple layers of features from tiny images. Technical report, University of Toronto.

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. Nature, 521(7553), 436-444.

Simonyan, K., & Zisserman, A. (2015). Very deep convolutional networks for large-scale image recognition. In 3rd International Conference on Learning Representations, ICLR 2015.

Smith, A. R. (1978). Color gamut transform pairs. ACM SIGGRAPH Computer Graphics, 12(3), 12-19.

Suzuki, S., & Abe, K. (1985). Topological structural analysis of digitized binary images by border following. Computer Vision, Graphics, and Image Processing, 30(1), 32-46.