Classification of COVID-19 based on CT images obtained from scientific papers.

Mikołaj Najda[1] and Dominik Ćwikowski[1]

[1]Wrocław University of Science and Technology
Faculty of Information and Communication Technology
276928@student.pwr.edu.pl[Mikołaj Najda]

Abstract. This project focuses on choosing the CNN model for COVID-19 classification from images and the dataset split in the context of a well-written machine learning project. Specifically, we compare the performance of DenseNet169, SimpleCNN, and EnhancedCNN models across two workflows (Workflow A - fixed data split and Workflow B - k-fold cross-validation). The evaluation is based on three metrics: F1-score, Accuracy, and AUC.
The dataset is preprocessed and a fixed number of training epochs are applied to each model. The models are then evaluated on the validation set for each workflow. Statistical tests, including the global and paired tests are conducted to analyze the significance of performance differences. The results reveal significant differences in performance between the models for all metrics in both workflows. DenseNet169 demonstrates superior performance compared to other models in workflow A, while SimpleCNN yields the best results in terms of workflow B.
These findings emphasize the importance of model selection in image classification tasks. The study highlights the need for careful consideration when choosing a model to maximize the performance of a given workflow, leaving space for hyperparameter tuning.

Keywords: COVID-19 · DenseNet169 · machine learning · classification.

1 Introduction

The COVID-19 outbreak has demonstrated the significant impact that the Internet of Things (IoT) and artificial intelligence (AI) fields have had on the healthcare industry, including the ability to assist with health monitoring, quarantine e-tracking, detection and diagnosis [1]. Rapid developments in machine learning have sparked new ideas among researchers [2]. AI-based models have been used to forecast the performance of illnesses by analyzing COVID-19-related symptoms, such as throat pain, immunity status, and diarrhoea [3]. Sentiment analysis of public opinions has also been performed based on tweets, using models such as SFODLD-SAC for data preprocessing and CRNN for sentiment analysis and classification [4]. Image classification, a task widely developed by deep learning researchers, has been used in conjunction with chest X-rays and computed tomography (CT) scans to detect lung changes caused by the virus. Machine learning techniques, such as Convolutional Neural Networks (CNNs), have shown their potential in these image-related solutions. The workflow for deploying deep learning-based tools typically includes gathering data, pre-processing (e.g. normalization, resizing, and segmentation), applying transfer learning or neural network architecture from scratch, setting up the classification classes based on the problem and data, and choosing appropriate metrics to evaluate model performance [5].

Convolution Neural Network (CNN) is the deep learning technique that enables achieving astonishing results in tasks related to image classification. It allows performing these tasks based on focusing on the relationship of the nearby pixels (contextual information). The general model of CNN is built with four components: convolution layer, pooling layer, activation function and fully connected layer [8]. The approach with CNN helps to avoid complicated feature engineering in medical image classification tasks. CNN-based transfer learning method seems to be one of the best choices for dealing with a small amount of data. By unfreezing the later layers, fine-tuning and omitting to overfit the model for a particular task is possible [9].

Transfer learning involves applying a pre-trained model on large datasets to a target model. Due to the lack of large medical image datasets which are properly labelled by professionals the transfer learning approach gained popularity. It allows for improving the generalization of the model, even if the dataset that the model was pre-trained on is not related of any kind to data for the particular problem. The pre-trained model is already learned to recognize features such as edges or corners which are common in every image. It is hard not to come across scientific papers about medical image classification where the authors did not try to apply this method [10].

Self-supervised learning is a form of semi-supervised learning, it tries to learn important correlations between extracted features of input unlabelled data, providing learning by solving auxiliary tasks. Pretext tasks are a type of auxiliary tasks which are performed to learn a model to extract useful features from images. Those problems may include predicting the missing part or rotation of an image. This approach allows using the large unlabelled datasets as a supportive way to pre-train the model and achieve better results on a target problem if only the pretext tasks are successfully designed [11].

The covid-19 outbreak gathered researchers who started developing deep learning tools in order to help diagnosticians with detecting lung pathologies caused by the virus from CT scans. A weakly-supervised framework for classification and lesion localization was presented. Weak supervision means that only image-level labels (virus positive or negative in this example) are used during training instead of pixel-level labels such as edges, and shapes of the region of interest (ROI). They have proposed an approach that contains three main steps: a feature extractor as a pre-trained and then fine-tuned CNN, a lesion localization module to identify potential lesion regions in the scans using a binary classifier and a fully connected network as a classification model. The framework achieved a better result (0.90) than classic approaches like RBF SVM, linear SVM or Random Forest, respectively (0.72, 0.75, 0.78) but give way to human expert (0.97). However, new approaches can enhance the classification tasks in the future and can become robust tools for professionals to improve their performance [16].

One group of researchers tried to develop the best tool for COVID-19 classification based on CT images. However, back in 2020, publicly available datasets were limited, so they gathered images from articles. The main question that arose was whether the images downloaded from the papers were still valuable in any pathology classification, given the loss in image quality and only selected slices. The authors of the challenge assured that the usability of the dataset was confirmed by an experienced radiologist who had been working with COVID-19 cases since the start of the pandemic [6]. They developed a sample-efficient deep learning method to demonstrate the usability of the low-sample dataset and besides big differences in target data (CT images) and data of pre-trained neural networks (animals, furniture) investigated transfer learning and self-supervised learning methods. In order to evaluate the classification task, they decided to test transfer learning based on pre-trained models on datasets such as ImageNet [18] and Lung Nodule Malignancy (LNM). A few different networks were applied to check the performance for the particular problem, the VGG16, ResNet18, ResNet50, DenseNet-121, DenseNet-169, EfficientNet-b0, and EfficientNet-b1 were evaluated. According to the knowledge of having a relatively small dataset, the light-weight neural network architecture was designed [7]. They used batch normalization, and binary cross-entropy as the loss-function, hyperparameters were tuned on the validation set as experimental settings and a different data augmentation for two approaches were implemented. They have shown results in three metrics: Accuracy, F1-score and AUC. They achieved the highest scores for DenseNet-169, F1-score and AUC metrics for the network that was pre-trained first on ImageNet, then on LMN respectively (0.82, 0.89), and the highest Accuracy (0.83) for the network that was pre-trained only on the ImageNet dataset. The self-supervised approach yielded the highest Accuracy, F1-score and AUC using the DenseNet-169, respectively (0.86, 0.85, 0.94) with the following steps: pre-train on ImageNet, perform Self-Supervised Learning (SSL) on Lung Nodule Analysis (LUNA) dataset without using labels of LUNA, then perform SSL on COVID19-CT without using labels of COVID19-CT and at the end fine-tune on COVID19-CT using labels. However, the authors evaluated the self-supervised learning using the data from the LUNA database and they did not include selected data in their GitHub, we took this fact into consideration during work with the problem and decided to... .

This paper presents two approaches to dealing with the dataset: one based on the authors' solution and the second on k-fold cross-validation to analyze the impact of choosing the preferred data split for n < 1000 [17]. Deep learning solutions were evaluated, all of the results are shown in this work. However, machine learning solutions in the medical imaging field continue to evolve, and each year new models and algorithms are developed. It is important to keep in mind that, especially in the healthcare industry, the accuracy of any solution must be high.

2 Proposed experimental design

2.1 Research questions

- Possibility of teaching the model medical image-related task using images obtained from scientific papers.
- Influence of the dataset split approaches on the models' performance.

2.2 Precise experiment purpose

The target of this work is to test the best deep learning model shown in [7], and compare its performance with the results given proposed, simple CNN. However, the test will be separated at the data preparation step. Once we test the data split proposed by the authors then we are going to do an approach with k-fold cross-validation. Ultimately, the results will be compared and discussed.

2.3 Dataset

The data for the training set were prepared as part of the Challenge - Grand Challenge on COVID-19 diagnosis from CT images. The images of COVID-19 positive were taken from scientific papers from medRxiv and bioRxiv publications, and the COVID-19 negative from sources such as the MedPix database and PubMed Central. Example CT scans from individuals who have been infected with the virus with visible lesions are shown in (Figure 2.3) and healthy individuals are presented in (Figure 2.3). COVID-related and non-COVID data information is presented in (Table 2.3).

Key notes about data:

- The corpus of test and validation images was procured from an external source featuring computed tomography (CT) images that were not acquired via paper downloads. Notwithstanding, the authors solely furnished the data distribution pertaining to the training, validation, and testing sets based on the images obtained from the papers, as demonstrated in Table 2.3. They were preventing the proper comparasion to their's results shown in the article.
- Utilizing a data split, as implemented by the authors, may not be optimal when dealing with a relatively small number of images (n < 1000), as suggested in [17]. Given that each data example is crucial for effectively training a model, we opted to utilize k-fold cross-validation instead of the holdout method in our approach, to mitigate the potential for model overfitting.

- Due to the small dataset, data augmentation has been performed using different random affine transformations such as random cropping with a scale of 0.5, horizontal flip as well as colour jittering with random contrast and random brightness with a factor of 0.2. However additional augmentations were added for the self-supervised approach.

Table 1. Dataset details.

COVID-19	Number
Positive	349
Negative	397

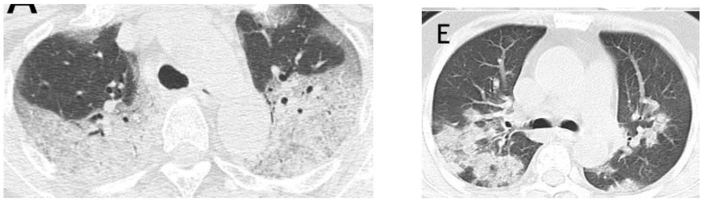


Fig. 1. CT COVID Positive.

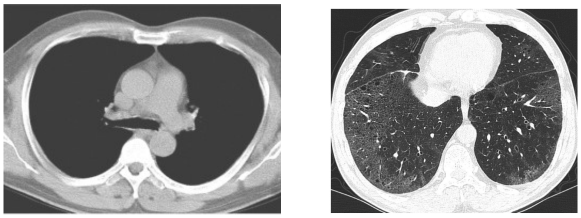


Fig. 2. CT COVID Negative.

Table 2. Author's images distribution.

Type	NonCOVID-19	COVID-19	Total
training	234	191	425
validation	58	60	118
test	105	98	203

2.4 Research protocol

1. Perform preprocessing on data and load dataset.
   The first step is to gather the data and prepare it for future models' learning. The data gathered by the authors of the challenge will be loaded using implemented data loader. The image transformation will be applied, it includes normalization, resizing and cropping to ensure that all the images are of the same size. Of course, images are going to be transformed into tensors in order to use the PyTorch library. However, further preprocessing might include data augmentation, if so the details will be provided.
2. K-fold cross-validation or authors' based dataset split.
   Due to lack of the data, the k-fold cross-validation is going to be applied in our approach compared to the authors' established in advance data split. K-fold cross-validation makes the most out of the available data and should help to obtain reliable performance estimates for models. By applying k-fold cross-validation, we can mitigate the impact of data deficiency and reduce the risk of overfitting or obtaining overly optimistic performance estimates.
3. Implementing the chosen models and the learning process.
   We are going to try and implement our own CNN, SimpleCNN architecture based on the authors' article and the best promising model from the article that we are focused on, which is DenseNet169 used with a transfer-learning approach. The proper training loop will be implemented.
4. Training the models.
   We will gather the best results and perform the metrics, such as accuracy, F1-score and AUC.
5. Statistics interpretation.
   The choice of statistical test depends on the nature of the data and the specific objectives of the analysis. Some common tests include t-tests, chi-square tests, or ANOVA. The specific test chosen will depend on factors like the number of groups to compare, the distribution of data, and whether the data is paired or independent. Analysis of the statistical results to determine if there are significant differences in performance between the models will be made, considering the p-values. If the differences are statistically significant, we can conclude that there is a notable distinction in performance between the models. However, if the differences are not statistically significant, we cannot confidently claim that one model is superior to the other based on the available evidence.

2.5 Description of the experimental environment

The project is written in Python programming language. PyTorch and scikit-learn (sklearn), widely used Python modules for machine learning, will be utilized for the development of the CNNs.

Table 3. Components description.

Component	Description
Graphics Card	NVIDIA GeForce GTX 1060
Memory	6GB
Processor	Intel(R) Core(TM) i5-7400
CPU	3.00GHz
RAM	16GB
Storage	1TB SSD

3 Methods and evaluation

3.1 Models

DenseNet169 is a convolutional neural network (CNN) architecture that belongs to the DenseNet family. It introduces the concept of dense connectivity, where each layer is directly connected to every other layer in a feed-forward fashion. DenseNet169 has 169 layers and utilizes a combination of convolutional, pooling, and dense (fully connected) layers. It has been widely used for tasks such as image classification and object detection. In this work, the model was pre-trained on the ImageNet dataset.

SimpleCNN (Fig. 5) consists of two main parts: a convolutional layer followed by a fully connected layer. The convolutional layer applies a set of filters to extract features from the input image. The ReLU activation function is applied to introduce non-linearity, and max pooling is used to downsample the feature maps. The output of the convolutional layer is then flattened and passed through a fully connected layer, which maps the extracted features to the desired number of classes. SimpleCNN is relatively straightforward and serves as a good starting point for learning and understanding CNN architectures.

EnhancedCNN (Fig. 5) refers to an improved version of the SimpleCNN architecture, which incorporates additional techniques to enhance its performance. These enhancements include batch normalization and dropout regularization. Batch normalization normalizes the inputs of each layer to reduce the internal covariate shift, which can improve the stability and speed of training. Dropout regularization randomly sets a fraction of the input units to zero during training, reducing overfitting and promoting better generalization. By incorporating these techniques, we aim to improve the model's ability to learn and generalize from the data.

3.2 Data transformations

These transformations are provided by the challenge authors and are performed as well in our data split approach. The transformations for the test set are the same as for the validation set. Images are also converted into PyTorch tensors.

Transformations:

- Resize: The images are resized to a height and width of 256 pixels while preserving the aspect ratio. Antialiasing is applied to smooth the edges during the resizing process.
- RandomResizedCrop: A random crop of size 224x224 is taken from the resized image. The scale parameter specifies the range of scales to randomly sample from, in this case, between 0.5 and 1.0. This helps in data augmentation and introduces diversity in the training set.
- RandomHorizontalFlip: The image is randomly flipped horizontally with a certain probability. This augmentation technique helps to increase the variability of the training data by providing different perspectives of the same object.

3.3 Workflow process

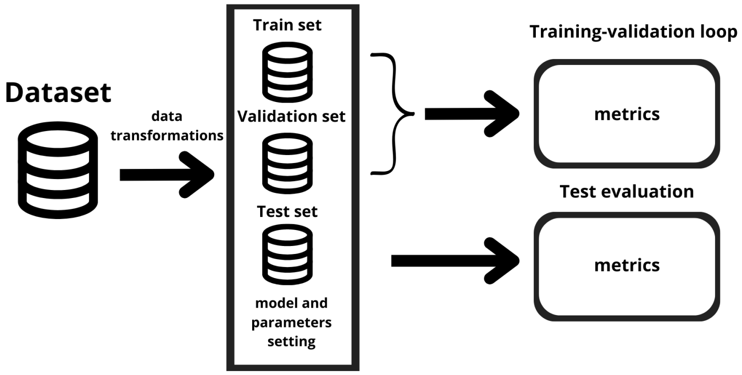


Fig. 3. Workflow A based on authors' data split.

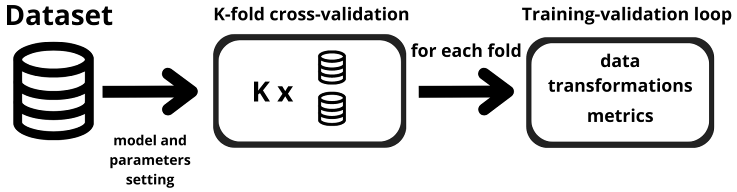
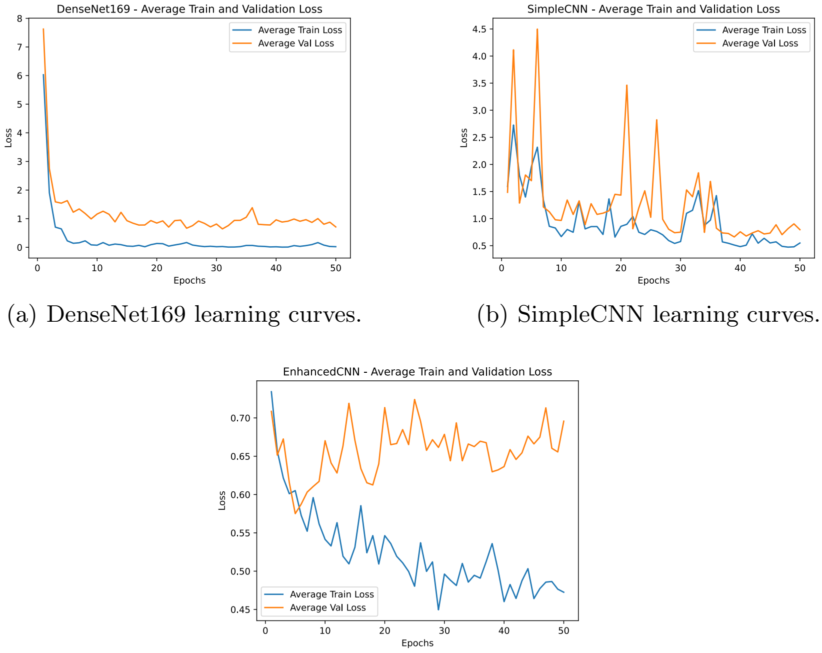


Fig. 4. Workflow B based on k-fold cross-validation.

4 Results

Table 4. Training parameters.

Parameter	Value
Learning rate	0.0001
Batch size	10
Number of epochs	50
Optimizer	Adam
Loss function	Cross Entropy
Optional (when k-fold cross-validation performed)	
K folds	3 or 5



(a) DenseNet169 learning curves.

(b) SimpleCNN learning curves.

(c) EnhancedCNN learning curves.

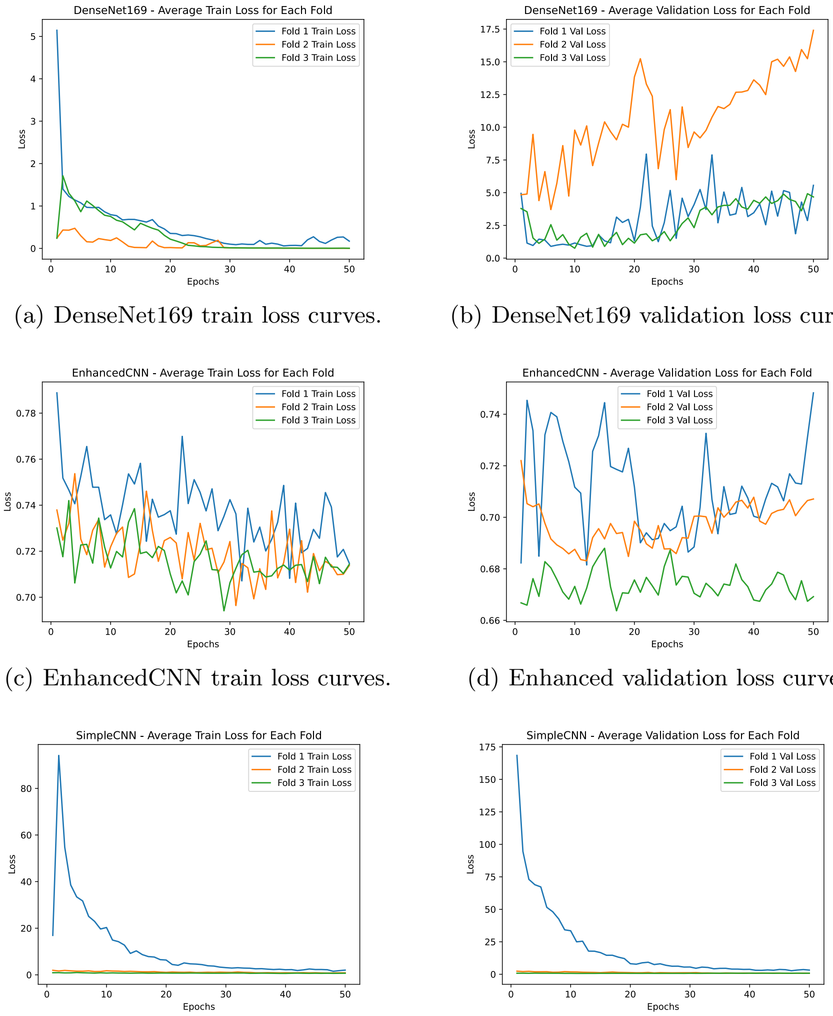
Fig. 5. Average Train Loss and Validation Loss curves for workflow A.

Diagnose the models' performance based on the article at website [19]:

(a) There is a minimal gap between those two curves and both of them continue to decrease which means that the model is learning well.

(b) The learning curves show noisy movement, there is a problem with the unrepresentative dataset for each, train and validation dataset.

(c) In an example of EnhancedCNN it is worth seeing that the validation learning curve is increasing when the train learning curve is decreasing with experience which suggests that the model is likely overfitting.



(a) DenseNet169 train loss curves.

(b) DenseNet169 validation loss curves.

(c) EnhancedCNN train loss curves.

(d) Enhanced validation loss curves.

(e) SimpleCNN train loss curves.

(f) SimpleCNN validation loss curves.

Fig. 6. Average Train Loss and Validation Loss for workflow B and 3 folds.

Diagnose the models' performance based on the article at website [19]:

(a) The training loss looks like a good fit for each fold.
(b) The validation loss curves show noisy movements for each fold which may mean that the validation dataset is not representative.
(c) The training dataset is unrepresentative and the train loss curves for each fold occur as noisy plots.
(d) The validation dataset is unrepresentative and the val loss curves for each fold occur as noisy plots.
(e) For the first fold the model looks like a good fit but in the next 2 folds seems like a underfit.
(f) For the first fold the model looks like a good fit but in the next 2 folds seems like a underfit. This problem is common when the model does not have a suitable capacity for the complexity of the dataset.

Table 5. Results for workflow A.

Model	Dataset	Accuracy	F1 score	AUC
DenseNet169	Validation (average results)	**75.28**	**0.76**	**0.75**
SimpleCNN		61.73	0.59	0.62
EnhancedCNN		65.23	0.65	0.65
DenseNet169	Test	**80.79**	**0.81**	**0.81**
SimpleCNN		61.57	0.61	0.62
EnhancedCNN		67.98	0.68	0.68

Table 6. Results for workflow B.

Model	Folds number	Avg. accuracy	Avg. F1 score	Avg. AUC
DenseNet169	3	50.569 %	0.376	0.477
SimpleCNN		**65.681** %	**0.609**	**0.646**
EnhancedCNN		53.425 %	0.480	0.517
DenseNet169	5	51.769 %	0.381	0.487
SimpleCNN		**71.009** %	**0.683**	**0.707**
EnhancedCNN		52.005 %	0.442	0.499

4.1 Statistics

All statistics tests assume that the significance level is equal to 0.05 and they are performed on validation results.

Table 7. Global tests results for workflow A.

Metric	Test	Result
f1-score	Shapiro-Wilk test (normality)	$p \leq 0.05$
	Levene's test (homogeneity of variance)	$p \leq 0.05$
	Kruskal-Wallis test	Statistic: 434.89, $p < 0.05$
	Significant difference between groups	Yes
accuracy	Shapiro-Wilk test (normality)	$p \leq 0.05$
	Levene's test (homogeneity of variance)	$p \leq 0.05$
	Kruskal-Wallis test	Statistic: 428.27, $p < 0.05$
	Significant difference between groups	Yes
auc	Shapiro-Wilk test (normality)	$p \leq 0.05$
	Levene's test (homogeneity of variance)	$p \leq 0.05$
	Kruskal-Wallis test	Statistic: 455.67, $p < 0.05$
	Significant difference between groups	Yes

Table 8. Global tests results for workflow B and k-fold = 5.

Metric	Test	Result
f1-score	Shapiro-Wilk test (normality)	$p \leq 0.05$
	Levene's test (homogeneity of variance)	$p \leq 0.05$
	Kruskal-Wallis test	Statistic: 228.72, $p < 0.05$
	Significant difference between groups	Yes
accuracy	Shapiro-Wilk test (normality)	$p \leq 0.05$
	Levene's test (homogeneity of variance)	$p \leq 0.05$
	Kruskal-Wallis test	Statistic: 223.84, $p < 0.05$
	Significant difference between groups	Yes
auc	Shapiro-Wilk test (normality)	$p \leq 0.05$
	Levene's test (homogeneity of variance)	$p \leq 0.05$
	Kruskal-Wallis test	Statistic: 252.95, $p < 0.05$
	Significant difference between groups	Yes

Table 9. Global tests results for workflow B and k-fold = 3.

Metric	Test	Result
f1-score	Shapiro-Wilk test (normality)	$p \leq 0.05$
	Levene's test (homogeneity of variance)	$p \leq 0.05$
	Kruskal-Wallis test	Statistic: 89.55, $p < 0.05$
	Significant difference between groups	Yes
accuracy	Shapiro-Wilk test (normality)	$p \leq 0.05$
	Levene's test (homogeneity of variance)	$p \leq 0.05$
	Kruskal-Wallis test	Statistic: 86.19, $p < 0.05$
	Significant difference between groups	Yes
auc	Shapiro-Wilk test (normality)	$p \leq 0.05$
	Levene's test (homogeneity of variance)	$p \leq 0.05$
	Kruskal-Wallis test	Statistic: 85.28, $p < 0.05$
	Significant difference between groups	Yes

Table 10. Paired Tests.

Metric	**Model**	**Shapiro-Wilk p-value**	**Result**
Workflow A vs Workflow B and k-fold = 5			
F1 Score	DenseNet169	$p \geq 0.05$	(T-Test) Significant Difference
	SimpleCNN	$p \geq 0.05$	(T-Test) Significant Difference
	EnhancedCNN	$p \geq 0.05$	(T-Test) Significant Difference
Val Acc	DenseNet169	$p \geq 0.05$	(T-Test) Significant Difference
	SimpleCNN	$p \geq 0.05$	(T-Test) Significant Difference
	EnhancedCNN	$p \geq 0.05$	(T-Test) Significant Difference
AUC	DenseNet169	$p \geq 0.05$	(T-Test) Significant Difference
	SimpleCNN	$p \geq 0.05$	(T-Test) Significant Difference
	EnhancedCNN	$p \geq 0.05$	(T-Test) Significant Difference

Table 11. Paired Tests.

Metric	Model	Shapiro-Wilk p-value	Result
Workflow A vs Workflow B and k-fold = 3			
F1 Score	DenseNet169	$p \geq 0.05$	(T-Test) Significant Difference
	SimpleCNN	$p \geq 0.05$	(T-Test) Significant Difference
	EnhancedCNN	$p \geq 0.05$	(T-Test) Significant Difference
Val Acc	DenseNet169	$p \geq 0.05$	(T-Test) Significant Difference
	SimpleCNN	$p \geq 0.05$	(T-Test) Significant Difference
	EnhancedCNN	$p \geq 0.05$	(T-Test) Significant Difference
AUC	DenseNet169	$p \geq 0.05$	(T-Test) Significant Difference
	SimpleCNN	$p \geq 0.05$	(T-Test) Significant Difference
	EnhancedCNN	$p \geq 0.05$	(T-Test) Significant Difference

Table 12. Statistical Test Results for workflow A.

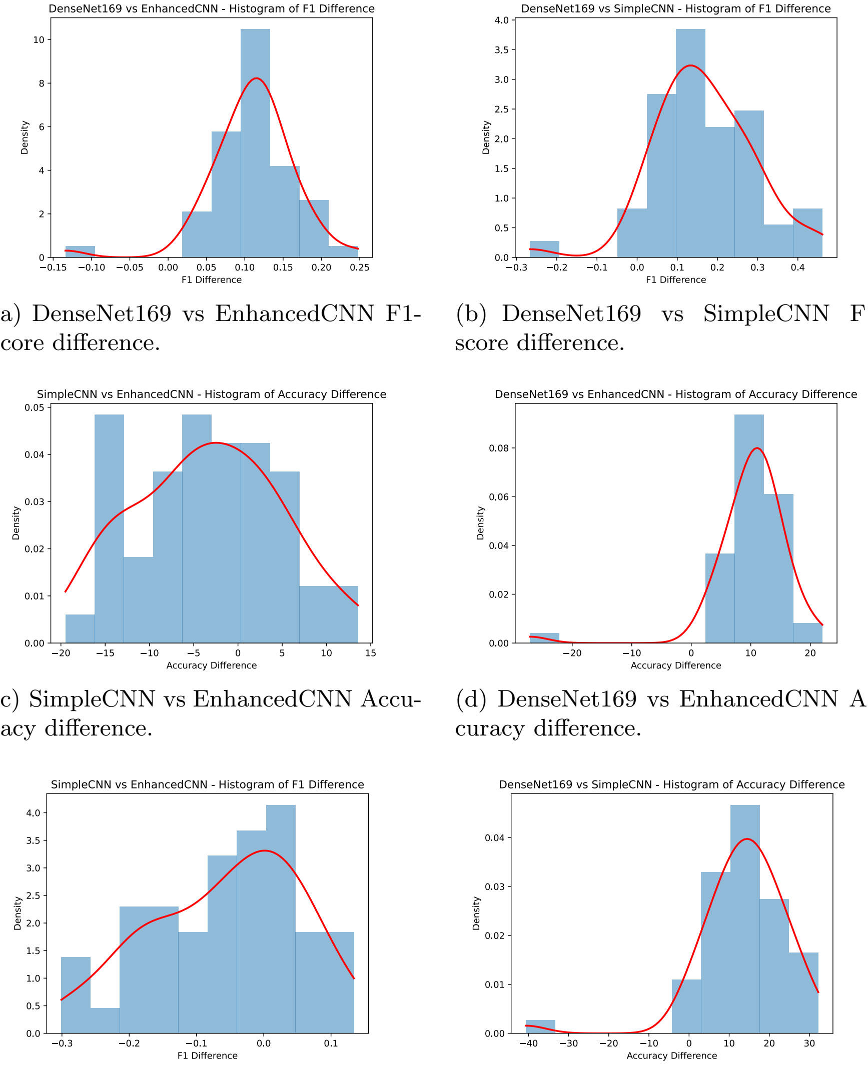
Metric	Models	Shapiro-Wilk p-value	Distribution	Test	Difference
F1-score	DenseNet169 vs SimpleCNN	$p \geq 0.05$	Normal	T-Test	Significant
	DenseNet169 vs EnhancedCNN	$p \leq 0.05$	Not normal	Wilcoxon	Significant
	EnhancedCNN vs SimpleCNN	$p \geq 0.05$	Normal	T-Test	Significant
Accuracy	DenseNet169 vs SimpleCNN	$p \leq 0.05$	Not normal	Wilcoxon	Significant
	DenseNet169 vs EnhancedCNN	$p \leq 0.05$	Not normal	Wilcoxon	Significant
	EnhancedCNN vs SimpleCNN	$p \geq 0.05$	Normal	T-Test	Significant
AUC	DenseNet169 vs SimpleCNN	$p \leq 0.05$	Not normal	Wilcoxon	Significant
	DenseNet169 vs EnhancedCNN	$p \geq 0.05$	Normal	T-Test	Significant
	EnhancedCNN vs SimpleCNN	$p \geq 0.05$	Normal	T-Test	Significant

Table 13. Statistical Test Results for workflow B and k-fold = 3.

Metric	Models	Shapiro-Wilk p-value	Distribution	Test	Difference
F1-score	DenseNet169 vs SimpleCNN	$p \leq 0.05$	Not normal	Wilcoxon	Significant
	DenseNet169 vs EnhancedCNN	$p \leq 0.05$	Not normal	Wilcoxon	Significant
	EnhancedCNN vs SimpleCNN	$p \leq 0.05$	Not normal	Wilcoxon	Significant
Val Acc	DenseNet169 vs SimpleCNN	$p \leq 0.05$	Not normal	Wilcoxon	Significant
	DenseNet169 vs EnhancedCNN	$p \geq 0.05$	Normal	T-Test	Significant
	EnhancedCNN vs SimpleCNN	$p \leq 0.05$	Not normal	Wilcoxon	Significant
AUC	DenseNet169 vs SimpleCNN	$p \leq 0.05$	Not normal	Wilcoxon	Significant
	DenseNet169 vs EnhancedCNN	$p \leq 0.05$	Not normal	Wilcoxon	Significant
	EnhancedCNN vs SimpleCNN	$p \leq 0.05$	Not normal	Wilcoxon	Significant

Table 14. Statistical Test Results for workflow B and k-fold = 5.

Metric	Models	Shapiro-Wilk p-value	Distribution	Test	Difference
F1-score	DenseNet169 vs SimpleCNN	$p \leq 0.05$	Not normal	Wilcoxon	Significant
	DenseNet169 vs EnhancedCNN	$p \geq 0.05$	Normal	T-Test	Significant
	EnhancedCNN vs SimpleCNN	$p \leq 0.05$	Not normal	Wilcoxon	Significant
Accuracy	DenseNet169 vs SimpleCNN	$p \leq 0.05$	Not normal	Wilcoxon	Significant
	DenseNet169 vs EnhancedCNN	$p \leq 0.05$	Not normal	Wilcoxon	Significant
	EnhancedCNN vs SimpleCNN	$p \geq 0.05$	Normal	T-Test	Significant
AUC	DenseNet169 vs SimpleCNN	$p \leq 0.05$	Not normal	Wilcoxon	Significant
	DenseNet169 vs EnhancedCNN	$p \geq 0.05$	Normal	T-Test	Significant
	EnhancedCNN vs SimpleCNN	$p \geq 0.05$	Normal	T-Test	Significant



(a) DenseNet169 vs EnhancedCNN F1-score difference.

(b) DenseNet169 vs SimpleCNN F1-score difference.

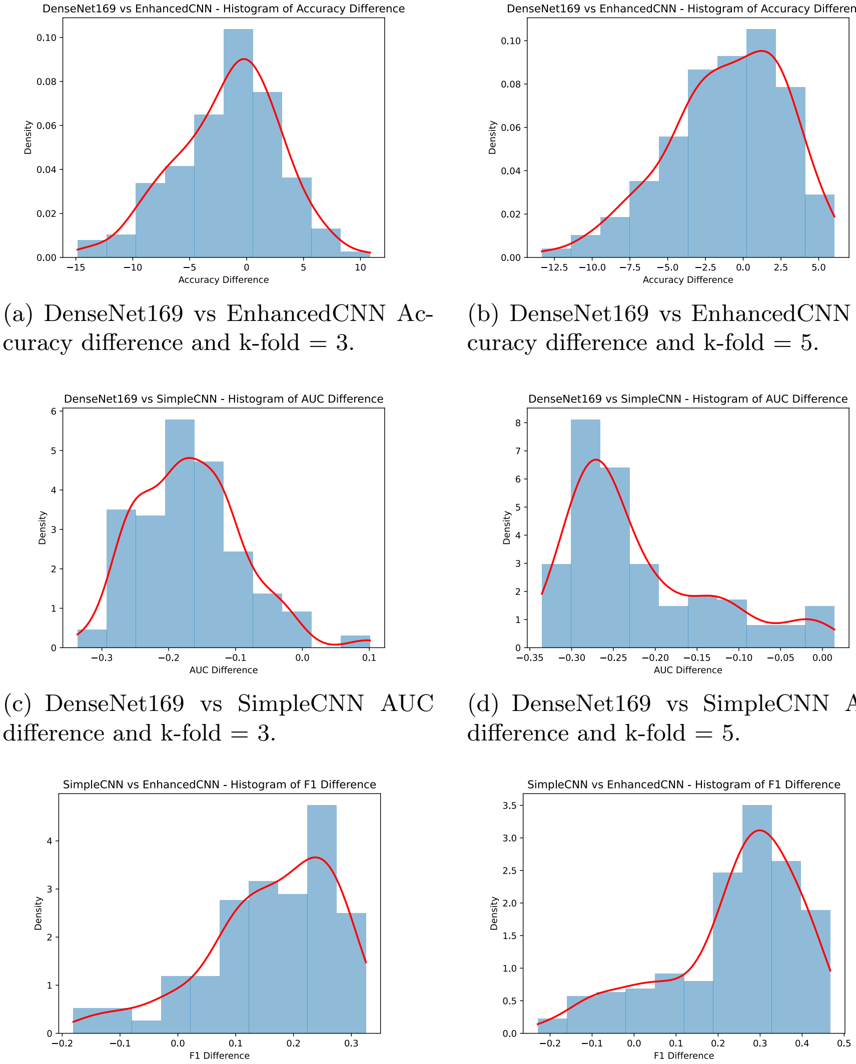
(c) SimpleCNN vs EnhancedCNN Accuracy difference.

(d) DenseNet169 vs EnhancedCNN Accuracy difference.

(e) SimpleCNN vs EnhancedCNN F1-score difference.

(f) DenseNet169 vs SimpleCNN Accuracy difference.

Fig. 7. Examples of distributions differences between values for workflow A.



(a) DenseNet169 vs EnhancedCNN Accuracy difference and k-fold = 3.

(b) DenseNet169 vs EnhancedCNN Accuracy difference and k-fold = 5.

(c) DenseNet169 vs SimpleCNN AUC difference and k-fold = 3.

(d) DenseNet169 vs SimpleCNN AUC difference and k-fold = 5.

(e) SimpleCNN vs EnhancedCNN F1-score difference and k-fold = 3.

(f) SimpleCNN vs EnhancedCNN F1-score difference and k-fold = 5.

Fig. 8. Examples of distributions differences between values for workflow B.

Both Workflow A and Workflow B show significant differences between the models for all metrics, indicating that the choice of the model significantly affects the performance. The data for all metrics in both workflows do not exactly follow a normal distribution, indicating the need for non-parametric tests like the Kruskal-Wallis test or Wilcoxon test. The unequal variance among the groups suggests that additional caution should be exercised when interpreting the results. The significant differences observed between the models in paired tests further support the notion that model selection plays a crucial role in performance.

In conclusion, based on the statistical test results, there are significant differences between the models for all metrics in both Workflow A and Workflow B. The choice of the specific test (T-test or Wilcoxon test) depends on the normality assumption of the data distribution. The results indicate that different models perform significantly differently from each other, suggesting that the choice of the model can have a significant impact on the performance of the workflow.

5 Summary

In this project, we have explored the topic of workflow optimization in the context of machine learning models. The main objective was to compare the performance of three different models (DenseNet169, SimpleCNN, and EnhancedCNN) across two different workflows (Workflow A and Workflow B). The evaluation of these models was based on three metrics: F1-score, Accuracy, and AUC.

Initially, we preprocessed the dataset and split it into training and validation sets using k-fold cross-validation. Then, we trained the models using a fixed number of epochs and recorded their performance on the training and validation set for each workflow which is shown on the learning curves plots.

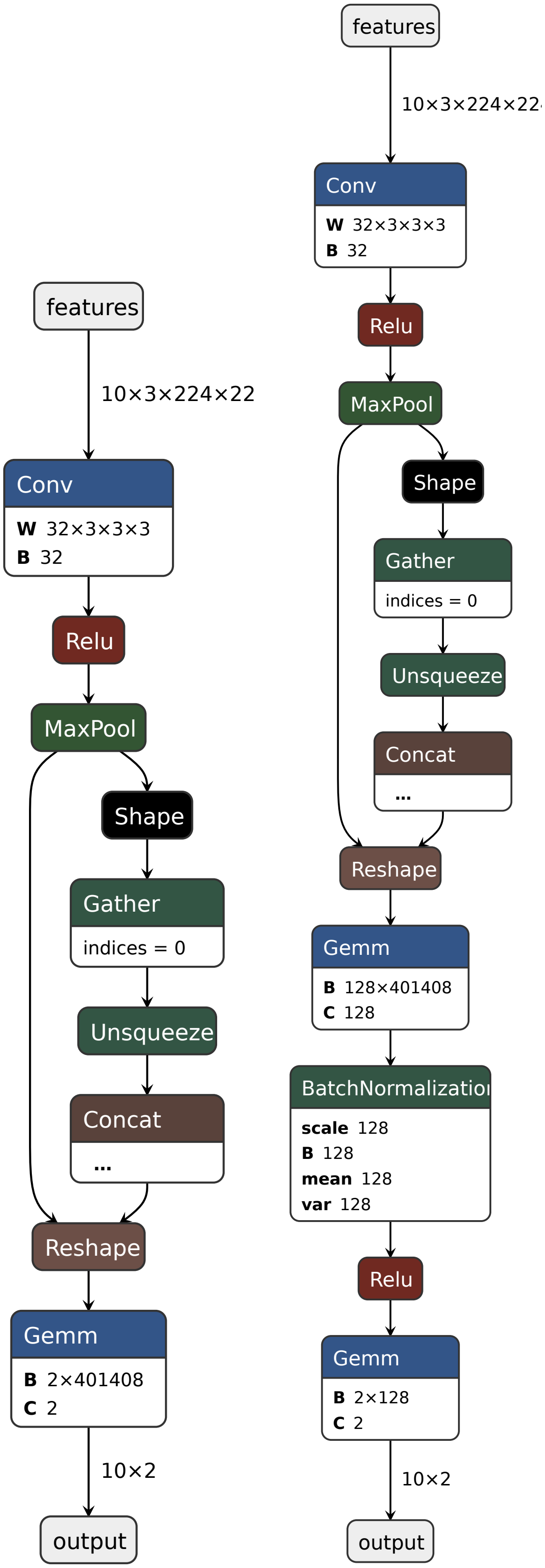
The analysis of the results revealed interesting insights. For Workflow A, the T-test and Wilcoxon test showed significant differences in performance between the models for the F1-score, Accuracy, and AUC metrics. In particular, DenseNet169 exhibited superior performance compared to other models in terms of the F1-score and AUC. For Workflow B, SimpleCNN indicated better performance in classification. This suggests that the choice of workflow has a substantial impact on the models' performance in terms of these metrics.

Overall, the statistical analysis highlights the importance of model selection in proposed workflows. The results demonstrate the significant differences in performance among the three models and emphasize the need for careful consideration when choosing the most suitable model for a given workflow.

References

1. Mondal S, Mitra P.: The Role of Emerging Technologies to Fight Against COVID-19 Pandemic: An Exploratory Review. Trans Indian Natl Acad Eng., 99–110 (2022)
2. Showmick Guha Paul, Arpa Saha, Al Amin Biswas, Md. Sabab Zulfiker, Mohammad Shamsul Arefin, Md. Mahfujur Rahman, Ahmed Wasif Reza: Combating Covid-19 using machine learning and deep learning: Applications, challenges, and future perspectives. Array, Volume 17 (2023)
3. Rehman MU, Shafique A, Khalid S, Driss M, Rubaiee S.: Future Forecasting of COVID-19: A Supervised Learning Approach. Sensors (Basel), (2021)

4. Alkhaldi, Nora A., Yousef Asiri, Aisha M. Mashraqi, Hanan T. Halawani, Sayed Abdel-Khalek, and Romany F. Mansour: Leveraging Tweets for Artificial Intelligence Driven Sentiment Analysis on the COVID-19 Pandemic. Healthcare 10, (2022)
5. Priya Aggarwal, Narendra Kumar Mishra, Binish Fatimah, Pushpendra Singh, Anubha Gupta, Shiv Dutt Joshi: COVID-19 image classification using deep learning: Advances, challenges and opportunities. Computers in Biology and Medicine, Volume 144 (2022)
6. Zhao, Jinyu and Zhang, Yichen and He, Xuehai and Xie, Pengtao: COVID-CT-Dataset: a CT scan dataset about COVID-19. arXiv preprint arXiv:2003.13865, (2020)
7. He, Xuehai and Yang, Xingyi and Zhang, Shanghang, and Zhao, Jinyu and Zhang, Yichen and Xing, Eric, and Xie, Pengtao: Sample-Efficient Deep Learning for COVID-19 Diagnosis Based on CT Scans. medrxiv, (2020)
8. Sakshi Indolia, Anil Kumar Goswami, S.P. Mishra, Pooja Asopa: Conceptual Understanding of Convolutional Neural Network- A Deep Learning Approach. Procedia Computer Science 132, 679–688 (2018)
9. Yadav, S.S., Jadhav, S.M.: Deep convolutional neural network based medical image classification for disease diagnosis. J Big Data 6, Volume 113 (2019)
10. Kevser Sahinbas, Ferhat Ozgur Catak: Transfer learning-based convolutional neural network for COVID-19 detection with X-ray images. Data Science for COVID-19, 451-466 (2021)
11. Kriti Ohri, Mukesh Kumar: Review on self-supervised image recognition using deep neural networks. Knowledge-Based Systems, Volume 224, (2021)
12. Nandi, Dibyadeep Ashour, Amira S. Samanta, Sourav Chakraborty, Sayan Salem, Mohammed Abdel-Megeed Mohammed Dey, Nilanjan: Principal Component Analysis in Medical Image Processing: A Study. International Journal of Image Mining, 45-64 (2015)
13. Mateen, Muhammad Wen, Junhao Nasrullah, Dr Song, Sun Huang, Zhouping: Fundus image classification using VGG-19 architecture with PCA and SVD. Symmetry, (2018)
14. Khachane, Monali: Organ-Based Medical Image Classification Using Support Vector Machine. International Journal of Synthetic Emotions, 18-30 (2017)
15. Jair Cervantes, Farid Garcia-Lamont, Lisbeth Rodríguez-Mazahua, Asdrubal Lopez: A comprehensive survey on support vector machine classification: Applications, challenges and trends, Neurocomputing, Volume 408, 189-215 (2020)
16. X. Wang et al.: A Weakly-Supervised Framework for COVID-19 Classification and Lesion Localization From Chest CT. IEEE Transactions on Medical Imaging, vol. 39, no. 8, pp. 2615-2625 (2020)
17. Baeldung on CS, https://www.baeldung.com/cs/train-test-datasets-ratio. Last accessed 19 March 2023
18. J. Deng, W. Dong, R. Socher, L. -J. Li, Kai Li and Li Fei-Fei: ImageNet: A large-scale hierarchical image database. IEEE Conference on Computer Vision and Pattern Recognition, (2009)
19. Machine Learning Mystery, https://machinelearningmastery.com/. Last accessed 28 May 2023



(a) SimpleCNN architecture.

(b) EnhancedCNN architecture.

Fig. 9. Models' architecture.