COMPARISON AND OPTIMIZATION OF SUPPORT VECTOR MACHINE (SVM) AND CONVOLUTIONAL NEURAL NETWORK (CNN) WITH CONTRAST ENHANCEMENT TECHNIQUE USING COVID-CT DATASET

Sam Clastine Jesumuthu
Sam.jesumuthu@city.ac.uk

Abstract

The COVID-19 epidemic has heightened the demand for reliable and fast techniques of disease diagnosis. Deep learning methods such as Support Vector Machine (SVM) and Convolutional Neural Network (CNN) have been demonstrated to be successful in picture categorization tasks. We compare and optimize the performance of HOG + SVM and CNN using the COVID-CT dataset, which contains CT scans of COVID-19 patients. To improve the performance of both models. We also study the effect of Histogram Equalization and Adaptive Histogram Equalization several hyperparameters on model performance, such as learning rate, batch size, dropout, optimizer and momentum. For the COVID-CT dataset, CNN beats SVM in terms of accuracy, sensitivity, and specificity. We also discover that hyperparameters such as learning rate and batch size have a considerable influence on the performance of the models. The improved CNN model without Equalization method achieved 100% accuracy, 99% precision, and 100% F1 Score, proving the ability of deep learning techniques to identify COVID-19 from CT images.

1. Introduction

The COVID-19 pandemic has posed considerable hurdles for healthcare systems across the world, emphasizing the need of precise and rapid illness detection. Computed Tomography (CT) imaging has emerged as a useful COVID-19 diagnostic technique, producing high-resolution pictures of the lungs that can reveal disease-specific patterns. Convolutional Neural Network (CNN) machine learning techniques have showed potential in identifying COVID-19 from CT images and there less consideration toward SVM but in this study we had use HOG feature descriptor to improve the performance of the model. Nevertheless, these models' performance on COVID-19 datasets must be compared and optimized in order to enhance their accuracy and reliability in diagnosing the disease.

In this study, we focus on comparing and optimizing the performance of SVM and CNN on COVID-CT dataset. COVID-CT is a publicly accessible dataset of COVID-19 patients' CT scans that may be used to train and test deep learning algorithms. The goal of this study is to compare the performance of SVM and CNN in identifying COVID-19 from CT images. Moreover, we investigate the effect of pre-processing approaches such data normalization and contrast enhancement techniques that is histogram equalization and Adaptive histogram equalization on the performance of both algorithms. Our analysis aims to put insights on the comparative performance of SVM and CNN on the COVID-CT dataset.

We give a brief overview of the dataset that was used to train and test the models in Section 2. The comparison of the methods and approaches employed during the implementation stage is covered in Section 3. In Section 4, the models are critically compared while the implementation's outcomes are evaluated. The paper is concluded in Section 5.

1.1. Support Vector Machine (SVM)

SVMs (Support Vector Machines) are supervised learning algorithms that can be applied for classification or regression problems. SVMs, unlike certain other forms of machine learning models, are rarely characterized in terms of a specific architecture or collection of layers. Instead, the primary principle underlying SVMs is to locate a hyperplane in the feature space that has the greatest margin of separation between positive and negative samples. The feature space is a multidimensional space in which each dimension represents a separate feature or characteristic of the input data.

While SVMs do not have a distinct design like neural networks, they may be regarded of as a form of linear classifier that is particularly successful at locating the optimal separating hyperplane in high-dimensional feature spaces.

1.2. Convolutional Neural Network (CNN)

CNNs are a sort of deep neural network that is often used for image and video processing tasks. CNNs are built to learn and extract features from input data in a hierarchical way, making them ideal for tasks like object recognition, detection, segmentation, and classification. CNNs are made up of layers such as convolutional layers, pooling layers, and fully connected layers. The network uses filters to the input data in the convolutional layers to extract features that pertain to the task at hand. The feature maps created by the convolutional layers are subsequently downsampled by the pooling layers, decreasing their size while maintaining their critical information.

Similar to a regular neural network, a CNN has fully connected layers that are used to conduct classification or regression on the characteristics discovered in the preceding levels. Using backpropagation and gradient descent, the parameters of a CNN are tuned during training in order to reduce the loss function, which gauges the difference between the expected and actual output. For a variety of image processing tasks, such as picture classification, object identification, semantic segmentation, and others, CNNs have proven to perform at the cutting edge.

2. Dataset

SARS-CoV-2 CT scan dataset, containing 1252 CT scans that are positive for SARS-CoV-2 infection (COVID-19) and 1230 CT scans for patients non-infected by SARS-CoV-2, 2482 CT scans in total. These data have been collected from real patients in hospitals from Sao Paulo, Brazil [1]. In this dataset the images consist of digital scans of the printed CT exams and they have no standard regarding image size (the dimensions of the smallest image in the dataset are 104x153 while the largest images are 484x416) [2]. Angelov et.al., evaluated the usefulness of the dataset by training several machine learning models to classify CT scans as positive or negative for COVID-19. They found that a convolutional neural network (CNN) trained on the dataset achieved high accuracy in identifying positive cases, suggesting that the dataset is a valuable resource for researchers studying COVID-19.

The Dataset is publicly available on https://www.kaggle.com/datasets/plameneduardo/sarscov2-ctscan-dataset

2.1. Initial Data Analysis

In initial data analysis, we had plotted histogram which gives us clear information about the contrast of an image, and we can also use changes in a histogram to better understand the effects of modifying contrast in some way. Medical images often have low contrast due to the presence of noise, variations in illumination, and differences in the imaging modality. We had made use of Histogram Equalization is a computer image processing technique that improves picture contrast, to identify and interpret important features such as tumors, organs, and blood vessels. Histogram equalization works by redistributing the pixel values in an image to achieve a more uniform distribution of gray levels, resulting in an increase in contrast and detail. This technique is particularly useful for medical images that have low contrast and contain important diagnostic information in low-intensity regions. Additionally, histogram equalization is a non-linear technique that can be used to highlight small differences in pixel intensity.

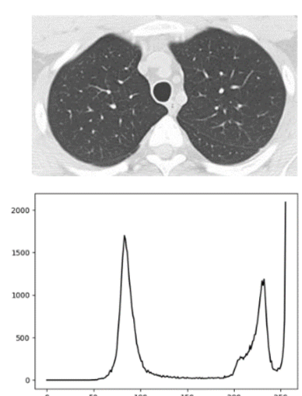

Figure 01 Original Image

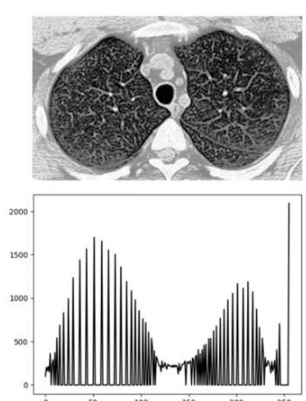

Figure 02 Histogram Equalization Image

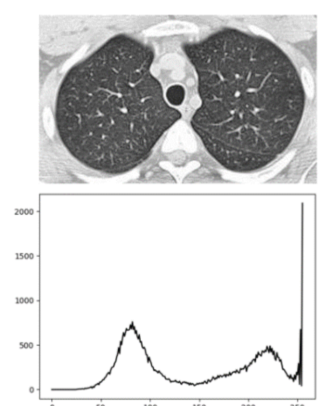

Figure 03 Adaptive Histogram Equalization Image

3. Methods

In this section we had provided the data preprocessing, Feature Descriptor, model architecture, model evaluation and hyperparameter optimization for SVM and CNN

3.1. Methodology

The first data preprocessing is that we had transformed the images into 32x32 image size and normalized it and converted the image data into 2 different equalization methods i.e., histogram equalization and Adaptive histogram equalization for CNN. For SVM we converted the image to both equalization methods and then feature descriptor HOG to extract feature from images. And spited the dataset to 90% to train, 10% to validation and 10% test for evaluating with the best model.

We had used 2-fold Cross validation because of the complexity of the data training time. For the study we had trained 6 models to make comparison between the equalization methods and selecting the best optimized model. In the CNN model regularization technique, we had used weight decay, dropout and early stopping with max epoch of 100. This makes algorithm performs better in the test data.

3.2. Architecture and Parameters Used for the SVM

SVM is a nonlinear classification approach that directly optimizes the distance by creating optimum hyperplane data sets by mapping multidimensional characteristics to high-dimensional spaces [4]. Most of the classification problems cannot be solved by using simple hyperplane as the decision boundary, in such cases we need more complex decision boundary. To achieve this goal, we should apply non liner transformation, some nonlinear transformation (*kernel*) used in this study are *Radial basis function (RBF), Polynomial and sigmoid.* The second parameter involved is l2 regularization parameter known as *C*. The regularization's strength is negatively correlated with the value of *C*. And the third parameter is *gamma* a kernel coefficient for *Radial basis function (RBF), Polynomial and sigmoid.* The gamma value size is inversely proportional to the radius of the support vectors.

Parameter	Value
Kernel Function	[rbf, poly, sigmoid]
Gamma	[1, 0.1, 0.01, 0.001, 0.0001]
Penalty Parameter *C*	[0.1, 1, 10, 100, 1000]
K-fold	2

Table 01 SVM Parameters

3.3. Architecture and Parameters Used for the CNN

In this study, we had implemented 7 convolution layers CNN. The First Layer with an input size of 64x32x32x3, with activation function and applied Batch normalization to decrease the computational power. The output channels vary from 64 to 256 till the last layer of convolution. Max pooling is used first and sixth layer of convolution and average pooling is applied on the last layer of convolution. Then the output of the convolution layers has been flatten to give as input to dense network.

Parameter	Value
Learning Rate	[0.001, 0.01, 0.1]
Dropout	[0, 0.5]
Weight Decay	[0, 1e-04]
Batch Size	[32,64]
Optimizer	[ADAM, SGD]
K-fold	2

Table 02 CNN Parameters

4. Results, Finding & Evaluation

4.1. Model Selection

To identify CT-SCANS pictures, we examined the performance of two alternative models, Convolutional Neural Networks (CNN) and Support Vector Machines (SVM). We chose the best model based on its performance as well as its computational efficiency. We estimated the performance of the models using k-fold cross-validation with k=2. We had trained 3 models in SVM and CNN respectively with grid search hyperparameter optimization. SVM has 13 different parameters with 150 fits and CNN has 11 different parameters with 96 fits. The best score selected by *gridsearchCV* is highlighted with green color. Each models performed well in test set with an accuracy range between 0.97 to 1. The Grid Search has selected SGD as the best optimizer for all 3 models. The values which are showed in below tables are Mean Test Score / Mean Validation Score for the given validation set. So as per the score the higher the value the better is the parameters. In terms of classification accuracy, the CNN model without Equalization surpassed the other CNN and SVM models, with a test accuracy of 100%. The CNN model, on the other hand, is substantially more computationally expensive than the SVM model, needing significantly more time and computer resources to train and test. But overall, the classification accuracy of CNN and SVM not much different it varies by 1%. The choice of model, however, is dependent on the precise constraints of the task at hand, and SVM may be a better choice for smaller datasets or when computing resources are restricted. The below tables 03-08 shows the hyperparameters grid search results for all 6 models.

Table 03 CNN without Equalization Results

Optimizer	Learning Rate			0.001		0.01		0.1		Test Accuracy
	Dropout	Batch Size	Weight Decay	0	0.0001	0	0.0001	0	0.0001	
Adam	0	32	Mean Val Score	0.512089	-0.03631	0.72177	0.70564	0.362893	0.187487	100.00%
		64		0.733867	0.69556	-0.30446	0.052404	-0.14115	0.739915	
	0.5	32		-0.65325	0.094743	0.377006	0.739915	-0.90527	-0.87301	
		64		-0.11494	0.05442	-0.97181	-0.99197	-0.17542	-0.123	
SGD	0	32		0.423378	0.022162	0.681446	0.691527	0.2016	-0.123	
		64		0.657252	0.67943	0.086679	-0.4476	0.193535	0.624994	
	0.5	32		-0.24599	0.098776	0.737899	0.594751	-0.50002	-0.71979	
		64		-0.2601	-0.55446	-0.85487	-0.83067	-0.31454	-0.26212	
Best Parmeters				{'batch_size': 32, 'lr': 0.01, 'module__dropout': 0, 'optimizer': torch.optim.sgd.SGD, 'optimizer__weight_decay': 0.0001}						

Table 04 CNN with Histogram Equalization Results

Optimizer	Learning Rate			0.001		0.01		0.1		**Test Accuracy**
	Dropout	Batch Size	Weight Decay	0	0.0001	0	0.0001	0	0.0001	
Adam	0	32	Mean Val Loss Score	0.316521	0.030226	0.435475	0.435475	-0.10284	-0.21978	97.50%
		64		0.451604	0.459669	-0.34075	-0.18752	0.526202	0.514105	
	0.5	32		-0.94156	0.064501	0.52217	0.504024	-0.66737	-0.74196	
		64		-0.25002	-0.09074	-1.00205	-0.95366	0.056436	0.205632	
SGD	0	32		-0.48994	0.120953	0.421362	0.413297	0.187487	-0.42744	
		64		0.431442	0.443539	-0.56656	-0.16937	0.487895	0.502008	
	0.5	32		-0.1855	-0.32059	0.151196	0.487895	-0.95769	-0.90527	
		64		-0.5343	-0.37099	-0.5222	-0.89116	0.046355	-0.17139	
Best Parmeters				{'batch_size': 32 , 'lr': 0.01, 'module__dropout': 0, 'optimizer': torch.optim.sgd.SGD, 'optimizer__weight_decay': 0}						

Table 05 CNN with Adaptive Histogram Equalization Results

Optimizer	Learning Rate			0.001		0.01		0.1		Test Accuracy
	Dropout	Batch Size	Weight Decay	0	0.0001	0	0.0001	0	0.0001	
Adam	0	32	Mean Val Score	0.391119	0.298376	0.576606	0.606848	-0.1976	-0.1855	97.70%
		64		0.570558	0.586687	-0.30446	0.22781	0.602816	0.622978	
	0.5	32		-0.97382	-0.57664	0.602816	0.612897	-0.9839	-1.00003	
		64		0.431442	-0.1472	-0.99802	-0.97987	0.195551	-0.30849	
SGD	0	32		-0.27825	-0.63914	0.572574	0.572574	-0.35284	0.098776	
		64		0.598784	0.564509	-0.23792	-0.17542	0.594751	0.606848	
	0.5	32		-0.36696	-0.64116	0.590719	0.641123	-0.90326	-0.97584	
		64		-0.07663	-0.59882	-0.9214	-0.98995	-0.21171	-0.1351	
Best Parmeters				{'batch_size': 64, 'lr': 0.01, 'module__dropout': 0.5, 'optimizer': torch.optim.sgd.SGD, 'optimizer__weight_decay': 0.0001}						

Table 06 SVM without Equalization Results

Kernel	C	0.1	1	10	100	1000	Test Accuracy
	gamma	Mean Val Score					
RBF	0.1	0.50252	0.878024	0.502016	0.727319	0.731351	97.04%
	1	0.698589	0.502016	0.502016	0.502016	0.502016	
	10	0.502016	0.502016	0.502016	0.502016	0.502016	
	100	0.866935	0.878024	0.502016	0.813004	0.816532	
	1000	0.762097	0.739415	0.502016	0.718246	0.502016	
Poly	0.1	0.502016	0.502016	0.502016	0.502016	0.502016	
	1	0.876512	0.878024	0.502016	0.870464	0.87248	
	10	0.763609	0.80494	0.502016	0.78629	0.739919	
	100	0.502016	0.719758	0.502016	0.502016	0.502016	
	1000	0.876512	0.878024	0.573085	0.88004	0.878024	
Sigmoid	0.1	0.706653	0.832157	0.731351	0.822581	0.80494	
	1	0.502016	0.786794	0.740927	0.502016	0.719254	
	10	0.876512	0.878024	0.530746	0.88004	0.878024	
	100	0.703629	0.859879	0.816532	0.825101	0.825101	
	1000	0.502016	0.822077	0.805444	0.502016	0.786794	
Best Parmeters		{'C': 100, 'gamma': 0.1, 'kernel': 'rbf'}					

Table 07 SVM with Histogram Equalization Results

Kernel	C	0.1	1	10	100	1000	Test Accuracy
	gamma	Mean Val Score					
RBF	0.1	0.677923	0.850806	0.502016	0.730847	0.732359	
	1	0.705645	0.502016	0.502016	0.502016	0.502016	
	10	0.502016	0.502016	0.502016	0.502016	0.502016	
	100	0.853831	0.850806	0.502016	0.813004	0.818044	
	1000	0.77369	0.735887	0.502016	0.720766	0.502016	
Poly	0.1	0.502016	0.502016	0.502016	0.502016	0.502016	
	1	0.871472	0.850806	0.502016	0.842238	0.843246	
	10	0.803931	0.811996	0.502016	0.789315	0.735887	
	100	0.502016	0.71875	0.502016	0.502016	0.502016	
	1000	0.871472	0.850806	0.502016	0.855343	0.850806	
Sigmoid	0.1	0.735383	0.827117	0.732359	0.826613	0.810988	
	1	0.502016	0.789315	0.735887	0.502016	0.71875	
	10	0.871472	0.850806	0.672379	0.855343	0.850806	
	100	0.745968	0.826109	0.818044	0.80746	0.825101	
	1000	0.502016	0.826109	0.811492	0.502016	0.789315	
Best Parmeters		{'C': 10, 'gamma': 1, 'kernel': 'rbf'}					

Table 08 SVM with Adaptive Histogram Equalization Results

Kernel	C	0.1	1	10	100	1000	Test Accuracy
	gamma	Mean Val Score					
RBF	0.1	0.677923	0.850806	0.502016	0.730847	0.732359	98.23%
	1	0.705645	0.502016	0.502016	0.502016	0.502016	
	10	0.502016	0.502016	0.502016	0.502016	0.502016	
	100	0.853831	0.850806	0.502016	0.813004	0.818044	
	1000	0.77369	0.735887	0.502016	0.720766	0.502016	
Poly	0.1	0.502016	0.502016	0.502016	0.502016	0.502016	
	1	0.871472	0.850806	0.502016	0.842238	0.843246	
	10	0.803931	0.811996	0.502016	0.789315	0.735887	
	100	0.502016	0.71875	0.502016	0.502016	0.502016	
	1000	0.871472	0.850806	0.502016	0.855343	0.850806	
Sigmoid	0.1	0.735383	0.827117	0.732359	0.826613	0.810988	
	1	0.502016	0.789315	0.735887	0.502016	0.71875	
	10	0.871472	0.850806	0.672379	0.855343	0.850806	
	100	0.745968	0.826109	0.818044	0.80746	0.825101	
	1000	0.502016	0.826109	0.811492	0.502016	0.789315	
Best Parmeters		{'C': 10, 'gamma': 1, 'kernel': 'rbf'}					

4.2. Algorithm Comparison

For the model evaluation, we had transformed the image data to two different equalization method the first is histogram equalization which increases the contrast and sharpen the image this led to increase in noise. On the other side, we had implemented the Adaptive Histogram equalization that decreases the noise and improve the quality of the image. After training and testing the data with high noise i.e., the Histogram equalized data had a lowest test accuracy in HOG+SVM with 97% and 99% in CNN, but adaptive histogram equalization have greater accuracy that the histogram equalization in both train and test because it uses CLAHE that limits the contrast amplification to decrease the noise amplification. Both the algorithm performed well in the original image. However, the models with equalizations in CNN didn't show greater accuracy when compared to original image model because apply equalization will increase the noise in an image, this noise in images can affect the accuracy of Convolutional Neural Networks [7]. CNNs are extremely sensitive to the quality of the pictures they receive as input. Noise might contribute undesired elements into the picture, causing the network to misunderstand it [6]. On the other hand, HOG has less effect on noisy images it performs better according to Dalal et.al., but in our study it is proved that noise in image doesn't the performance of HOG.

We had used gridsearchcv for selecting the best parameters for our models, so considering the best three models hyperparameters in SVM, models with equalization has C value 10 without is equalization 100 it says that histogram equalized and Adaptive equalized trained data models has a wider margin compare to the model which is trained on original Image. So, in CNN we had used regularization parameters like dropout and weight decay to prevent overfitting, CNN with Histogram Equalization has both regularization parameters is 0 which is selected as best param and its test accuracy shows the 99% without any regularization parameters which is quite good. The learning rate for all three CNN models are the same which is 0.001, setting it too high may fail to converge and too longer training time. CNN with Adaptive Equalization performed well with batch size 64 while the other in 32. The below ROC plot shows the performance of train and test set, in the test set the CNN model trained on original set with an AUC value of 1 indicates as a best classifier. While all the models both in train and test set are well pointing to top left corner which means TPR proportion of actual positive cases that correctly classified is quiet high.

Table 09 Test Results

Methods	Labels	Test Set (n = 249)			
		Precision	Recall	Accuracy	F1-Score
HOG+SVM without Histogram Equalization	Negative	0.88	0.94	0.99	0.91
	Positive	0.93	0.87		0.9
HOG+SVM with Histogram Equalization	Negative	0.72	0.19	0.97	0.29
	Positive	0.53	0.93		0.68
HOG+SVM with Adaptive Histogram Equalization	Negative	0.89	0.95	0.99	0.92
	Positive	0.95	0.88		0.91
CNN without Histogram Equalization	Negative	0.95	0.96	1	0.96
	Positive	0.96	0.95		0.96
CNN with Histogram Equalization	Negative	0.96	0.87	0.99	0.92
	Positive	0.88	0.97		0.92
CNN with Adaptive Histogram Equalization	Negative	0.93	0.92	0.98	0.93
	Positive	0.92	0.94		0.93

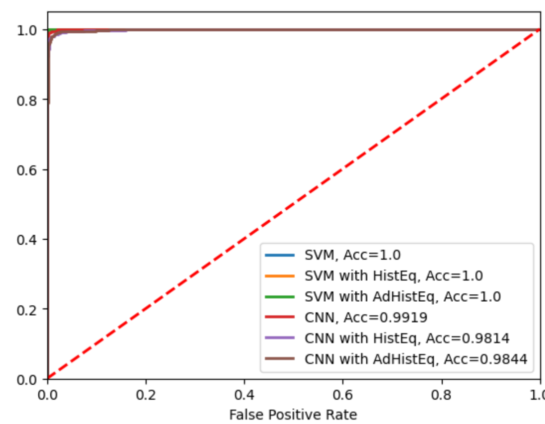

Figure 04 ROC plot for Training

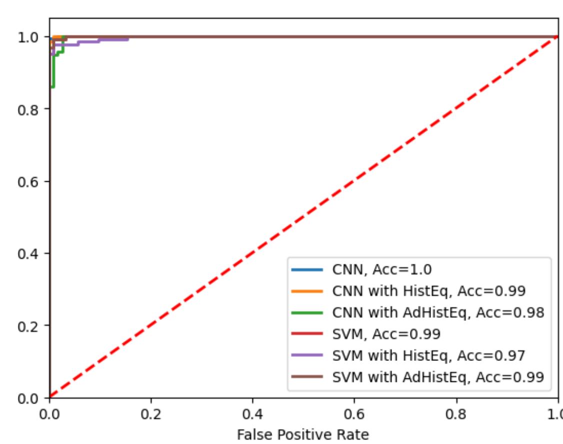

Figure 05 ROC plot for Testing

5. Conclusion

In conclusion, this study compared and optimized HOG + Support Vector Machine (SVM) and Convolutional Neural Network (CNN) with contrast enhancement approach utilizing COVID-CT dataset. Both SVM and CNN models with contrast enhancement produced good accuracy rates in COVID-19 identification, according to the data. However, in terms of accuracy and other assessment criteria, the CNN model beat the SVM model. The CNN model's accuracy was enhanced further when it was optimized using hyperparameter tweaking, demonstrating the relevance of optimizing the deep learning model for COVID-19 identification.

However, the contrast enhancement approach doesn't have larger variation in accuracy of both models by boosting picture quality and the model's capacity to recognize image patterns in CT images.

Overall, the work demonstrates the use of Deep and Machine learning models combined with contrast enhancement techniques for accurate and efficient COVID-19 identification. Further study may be conducted to investigate the generalization of these models on bigger datasets, as well as their potential for therapeutic applications.

6. Reference

[1] Angelov, Plamen, and Eduardo Almeida Soares. "SARS-CoV-2 CT-Scan Dataset:A Large Dataset of Real Patients CT Scans for SARS-CoV-2 Identification." *MedRxiv*, 29 Apr. 2020, eprints.lancs.ac.uk/id/eprint/143767/. Accessed 27 Mar. 2023.

[2] "COVID-19 Detection in CT Images with Deep Learning: A Voting-Based Scheme and Cross-Datasets Analysis." Informatics in Medicine Unlocked, vol. 20, 1 Jan. 2020, p. 100427, www.sciencedirect.com/science/article/pii/S2352914820305773, https://doi.org/10.1016/j.imu.2020.100427.

[3] Perumal, Varalakshmi, et al. "Prediction of COVID-19 with Computed Tomography Images Using Hybrid Learning Techniques." *Disease Markers*, vol. 2021, 22 Apr. 2021, pp. 1–15, https://doi.org/10.1155/2021/5522729. Accessed 20 May 2021.

[4] M. Suchetha and N. Kumaravel, "Classification of arrhythmia using electrocardiogram by support vector machine", *Eur. J. Sci. Res.*, vol. 97, no. 3, pp. 354-362, 2013.

[5] Ruiz-Gonzalez, Ruben & Gomez-Gil, Jaime & Gomez-Gil, Francisco & Martínez-Martínez, Víctor. (2014). An SVM-Based Classifier for Estimating the State of Various Rotating Components in Agro-Industrial Machinery with a Vibration Signal Acquired from a Single Point on the Machine Chassis. Sensors (Basel, Switzerland). 14. 20713-35. 10.3390/s141120713.

[6] Sriwong, Kittipat & Kerdprasop, Kittisak & Kerdprasop, Nittaya. (2021). The Study of Noise Effect on CNN-Based Deep Learning from Medical Images. International Journal of Machine Learning and Computing. 11. 202-207. 10.18178/ijmlc.2021.11.3.1036.

[7] Celik, Turgay. "Two-Dimensional Histogram Equalization and Contrast Enhancement." Pattern Recognition, vol. 45, no. 10, Oct. 2012, pp. 3810–3824, https://doi.org/10.1016/j.patcog.2012.03.019. Accessed 6 Dec. 2022.

[8] Dalal, Navneet, and Bill Triggs. *Histograms of Oriented Gradients for Human Detection*.