IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received January 27, 2022, accepted February 21, 2022, date of publication February 24, 2022, date of current version March 8, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3154405

SVMnet: Non-Parametric Image Classification Based on Convolutional Ensembles of Support Vector Machines for Small Training Sets

HUNTER GODDARD AND LIOR SHAMIR
Department of Computer Science, Kansas State University, Manhattan, KS 66506, USA
Corresponding author: Hunter Goddard (hbgoddard@ksu.edu)

This work was supported by NSF under Grant AST-1903823.

ABSTRACT While deep convolutional neural networks (DCNNs) have demonstrated superiority in their ability to classify image data, one of the primary downsides of DCNNs is that their training normally requires large sets of labeled ''ground truth'' images. For that reason, DCNNs do not provide an effective solution in many real-world problems in which large sets of labeled images are not available. Here we propose to use the quick learning of SVMs to provide a solution for learning from small image datasets in a non-parametric manner. Experimental results show that while ''conventional'' DCNN architectures such as ResNet-50 outperform SVMnet when the size of the training set is large, SVMnet provides a much higher accuracy when the number of ''ground truth'' training samples is small.

INDEX TERMS Artificial neural networks, image classification, machine learning, support vector machines.

I. INTRODUCTION

Deep convolutional neural networks (DCNNs) are powerful tools for multiple tasks of automatic image analysis, demonstrating paramount success and consequently gaining substantial popularity over the past decade. By analyzing the pixels directly, CNNs can be applied to various types of image content without the need to develop task-specific algorithms, and can easily be applied to a broad range of domains with excellent performance [1].

One of the major weaknesses of modern DCNNs is their dependence on a large set of examples for training. Cutting-edge DCNNs can have hundreds of layers, each with thousands of trainable parameters. For instance, the common ResNet-50 [2] contains over $2 \cdot 10^6$ artificial neurons. Therefore, to achieve meaningful performance and avoid overfitting, DCNNs normally rely on relatively large training sets.

Training DCNNs normally requires large datasets of labeled ground truth images. Commonly used datasets include benchmarks such as ImageNet or the Modified National Institute of Standards and Technology (MNIST) dataset of handwritten characters. These benchmark datasets provide tens of thousands of images with high-quality annotations for training deep CNNs, and are commonly used for testing their performance. However, in many cases of real-world image classification problems, large datasets of clean, labeled ground truth are not available.

For instance, in the biomedical domain machine learning is often used for the purpose of image-based diagnostics [3]. However, the acquisition and annotation of each image can require the use of costly medical instrumentation, technicians, and medical staff who can annotate each sample manually [4], [5]. Acquiring a single MRI image can take 30 minutes or more of using the instrument, excluding the time required to prepare the subject. The cost involved in the acquisition of such image is non-negligible. Even when using a quicker and less expensive imaging such as x-rays, the annotation of the data normally requires two or more trained experts, and the time they invest in the annotation is both expensive and time-demanding. That bottleneck has substantial impact on the ability of researchers in the medical domain to acquire large datasets.

Additionally, in the biomedical domain, human protection procedures and protocols are required for the acquisition of each sample, making the preparation of large datasets less practical. Therefore, biomedical image datasets are normally far smaller than the modern datasets commonly used to train DCNNs such as MNIST or ImageNet.

Rare cases can also make it difficult to acquire a suitable training set [6]. For instance, to prepare an image-based

The associate editor coordinating the review of this manuscript and approving it for publication was Yudong Zhang.

diagnostics system that can automatically detect a rare clinical condition, a sufficiently large number of images of that rare case is required. In many cases, even when the resources are not limited by neither time nor cost, a sufficiently large number of cases is difficult to find.

In some cases the acquisition of images can involve substantial pre-processing, preparation of slides, staining, and imaging of each slide [7]. That is often the case when performing histological analysis for the purpose of diagnostics using machine learning [8]–[10].

Clearly, situations in which the dataset is small are not limited to the biomedical domain. Scientific experiments that require annotated data are very often limited by the resources required to annotate them. One of the solutions that the scientific community proposed is the use of crowdsourcing [11]–[13]. By crowdsourcing, non-expert volunteers can help annotating images or other data. With a large number of volunteers, the annotation of large datasets becomes feasible, and the resulting annotated datasets can be used to train machine learning systems. However, such crowdsourcing campaigns can take several years to complete [14], and are subjected to human error and human perceptional bias [15]. In many cases the annotation requires an expert, and the task cannot therefore be performed by anonymous untrained volunteers. In practice, experimentalists are often limited in their ability to utilize crowdsourcing for annotating a specific dataset.

The need for a large number of training samples is a practical downside of DCNNs, making them difficult to use optimally in many real-world cases. A common solution to increasing the size of the training set is data augmentation, in which different modifications of the images in the original dataset can create more training samples. However, that strategy can also lead to biases by overusing the same examples. In some cases transfer learning can be used to fine-tune neural networks using pre-trained models. Transfer learning is a proven tool to reduce the required training set size, but for domains with very small datasets for fine-tuning, the pre-trained models may remain too sensitive to their original task.

The problem of small training sets has been addressed in the past by using previous knowledge for few-shot training [6] and even one-shot training [16]–[20]. These methods reduce the number of required samples dramatically to as low as just one, but also require prior knowledge that is not necessarily available in all cases. Other related solutions include 3-D octave convolution with the spatial-spectral attention network [21] or deep attention graphs [22] for the problem of hyperspectral image classification.

This paper explores a new form of non-parametric image classification in cases when the number of samples is limited. Based on an ensemble composition of support vector machines (SVMs), the method can work with no prior knowledge, in a similar manner to ''standard'' supervised machine learning. Inspired by CNN architecture, SVMnet utilizes a large number of small SVMs to quickly analyze image patches, structured in layers that allow for stacking or custom ensemble techniques. An SVM [23] is less sensitive to high-dimensionality feature spaces [24]–[26], and can learn from a relatively small number of training samples [27]–[30] compared to other supervised machine learning approaches.

The primary advantage of the proposed method is that it outperforms the common DCNN architectures in cases when the number of labeled training images is small. As discussed above, such cases are not uncommon in real-world settings. Another advantage of the method is its much shorter training time compared to the time required to train deep neural networks.

II. ARCHITECTURE OF SVMNET

The proposed SVMnet architecture is designed as a stacked ensemble of numerous simple SVM classifiers organized into one or more layers. Each layer is an array of SVMs which functions similarly to a convolutional layer in a CNN. Each SVM in a layer is independent and all are assigned an equal-sized patch of the layer's input, referred to as a window. Variable stride length and padding, as described in Chapter 2 of [31], are specified as hyperparameters. Each input to the following layer is the output of one SVM.

When a layer is evaluated, each SVM in the layer is trained on ground truth labels. The input to the SVM is the flattened portion of each input image that is within the SVM's window. Each pixel channel within the window is essentially treated as one input feature. For instance, a 5×5 window would create a 25-feature SVM for grayscale input and a 75-feature SVM for 3-channel RGB input. During this step, the SVMs may be given weights based on the accuracy of the fit, used for ensemble classification. Each SVM then predicts a class label or a vector of class probabilities for its window of each input, creating an input tensor for the next layer.

Fig. 1 shows a simple layer in SVMnet. Each node in the layer is one SVM, trained using the ground truth labels for the input samples. The weights are determined based on the classification accuracy of the SVM compared to the ground truth of the training set. The weight function is configurable and will be described later in this Section.

To produce one class label for each input, SVMnet may perform a weighted vote after the final layer. This vote combines the results of the final layer by treating each value as a vote for that class label. If the final layer is weighted, these are used to weigh the votes in favor of SVMs with higher accuracy.

$$S_c = \Sigma_i \eta(A_i)[P_i = c] \tag{1}$$

The total voting score S_c of each class c is calculated by (1), where A_i is the accuracy score of SVM i in the final layer, η is the weight function, and P_i is the class label predicted by SVM i. That is, if the predicted label P_i of SVM i is class c, the weighted score $\eta(A_i)$ is added to the vote for that class. The weight function emphasizes the predictions of the SVMs with higher accuracy during training. The class that has the highest score S_c is chosen as the predicted label by the model for the given sample. The weight function η is configurable

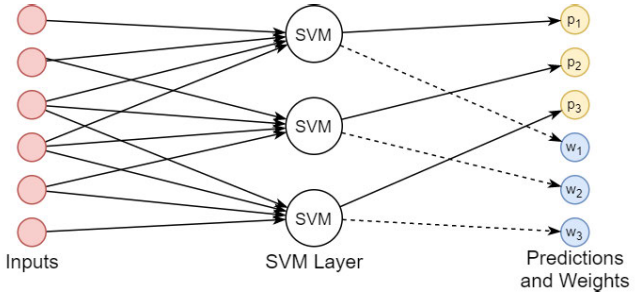

FIGURE 1. Example of a simple weighted layer of SVMnet. Each node in the layer is an SVM, trained with a subset of the inputs (pixels). Weight outputs are optional for a given layer.

and in our experiments is defined as $\eta(x) = x^2$, where x is the classification accuracy of the SVM determined during training.

While the layers support arbitrary estimators, here we use only support vector machines (SVM), hence the name SVMnet. The SVMs are trained with a Radial Basis Function (RBF) kernel [32] and scaling gamma value, and they continue to iterate until convergence with a 0.001 tolerance. The ability to choose different estimators in each layer can be compared to the ability to use different activation functions in the layers of neural networks.

Fig. 2 illustrates one possible two-layer SVMnet architecture. Each SVM in the first layer analyzes a specific patch of each image and is fitted independently against ground truth labels. These SVMs then produce a vector of class probabilities for the same pixel region which forms the input matrix for the following layer. The SVMs in the second layer are fitted on a region of these probabilities and predict a class label for the image. These labels are then tallied in a final vote to produce one label for the input. The motivation for multiple layers is that layers after the first can in essence learn which of the SVMs in their window are more accurate or ''trustworthy'', as their predictions are being compared to ground truth labels in each layer.

A. DROPOUT

Not every patch is expected to produce a well-informed SVM. Some regions of the images, particularly towards the edge, often lack the details necessary to distinguish samples from each other. This can cause the outputs of these SVMs to act as noise in a vote tally. Even with the expected low accuracy score of the SVM depressing the weight of its vote, if the low-information regions are large then enough inaccurate votes may overwhelm the more accurate votes. To help prevent this, a dropout system is implemented for the vote tally.

When using dropout, which SVMs to drop are calculated when fitting SVMnet. First, the SVMs are ordered from the highest weight to the lowest. Votes are then cumulatively tallied one SVM at a time with the accuracy of the votes measured between each tally. SVMnet then finds the global maximum accuracy of the cumulative tally. This marks the point where including the votes of the less-accurate SVMs lowers the overall accuracy of the tally, so those SVMs are marked for dropout and are not included in the final vote. When the model is used to make predictions, the vote will only include the outputs of the SVMs that contributed to the most accurate tally.

In most cases during testing, automatic dropout resulted in equal or better performance than without dropout, as the least informative regions of the image were ignored. However, as with all hyperparameters, performance sometimes decreased and required fine-tuning. In each of the experiments described in Section III, the SVMnet model presented is the one with the highest-performing hyperparameters among the combinations tested.

B. FORMAL DEFINITION OF SVMNET

SVMnet can be defined formally as a 4-tuple as shown by Equation 2:

$$SVMnet = (T, C, S_0, \Phi), \tag{2}$$

where T is the topology of the network, C is the initial constants, S_o is the initial state of the network, and Φ is the set of SVM classifiers. The components that make the SVMnet are defined by Equation 3.

$$\begin{aligned} T &= (V, E) \\ C &= \{W, \Theta\} \\ S_0 &= \{\psi_i\} \\ \Phi &= \{(\Xi_i, \gamma_i, C_i)\} \end{aligned} \tag{3}$$

The topology $T = (V, E)$ reflects the structure of the network, where V is the nodes and E is a set of connections $E_{i,j}$ between the nodes $V_i \rightarrow V_j$, where V_i and V_j are two connected nodes. A pair of nodes $V_i, V_j \in V$ can have one or zero connections between them. Like in artificial neural networks, the topology $T = (V, E)$ determines the number of layers, number of nodes per layer, and the kernel size.

The constants C include the thresholds W, which are the threshold values used for ignoring the output of an SVM classifier as explained in Section II-A. Each connection $E_{i,j}$ between two nodes is assigned with a threshold $W_{i,j}$, which determines whether the output of the SVM node i is used as an input to SVM node j. Unlike neural networks, in SVMnet these threshold values are constants, as they are not changed during training. Whether these threshold values impact the analysis depends on the consistency of the input, such that an inconsistent SVM node is ignored if its consistency observed using the ground truth training data does not meet the threshold. The use of these thresholds is explained in detail in Section II-A. Another constant is Θ, which is the number of classes.

The initial status of the network S_0 is a collection of SVM hyperplanes ψ, such that the hyperplane ψ_i is the initial hyperplane of the SVM in node i. The hyperplanes are changed during the training of the SVMnet, as the SVMs learn from the data.

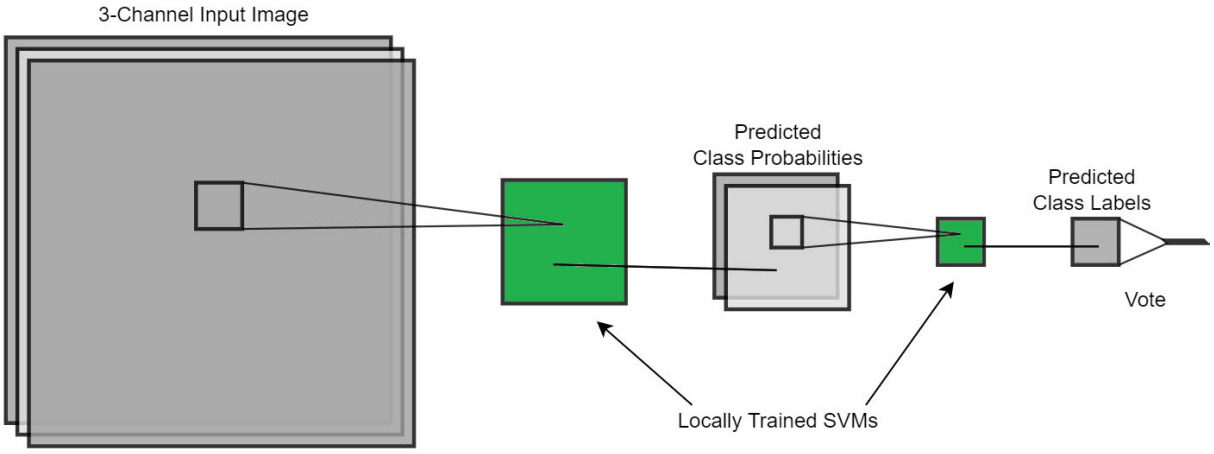

FIGURE 2. Example SVMnet architecture containing two SVM layers (in green) and a class label vote. Each SVM is trained on a patch of the layer's input. An $n \times m$ SVM layer produces $n \times m \times d$ output ($d \geq 1$).

The set of SVM classifiers Φ is defined by $\{(\Xi_i, \gamma_i, C_i)\}$, such that each SVM classifier Φ_i is defined by its kernel Ξ_i, its gamma parameter γ_i, and its C parameter C_i. In the implementation shown in this paper all SVMs are defined by the same parameters, but other implementations are also possible in which different SVMs have different kernels or other parameters.

III. EXPERIMENTAL RESULTS

To test the efficacy of SVMnet compared to a ''conventional'' CNN, several experiments were performed using common, relatively small datasets. The purpose of SVMnet is not to outperform CNNs in the general case, but to achieve higher accuracy when the number of labeled training images is limited. Therefore, the experiments were made with different sizes of training sets to compare the classification accuracy as the training set increases.

The performance of the SVMnet was compared to the performance of residual network, or ResNet, models with 18, 34, and 50 layers [2]. ResNet is a powerful architecture that was designed to reduce the number of required training samples for deep learning tasks and has demonstrated excellent efficacy in image classification. Each ResNet model was compared when trained from scratch and when fine-tuned using pretrained ImageNet weights. Following the practice in [2], the final convolutional layer is followed by a global average pooling layer, then by a single fully-connected layer with softmax activation and as many units as class labels in the respective task. Models were trained using stochastic gradient descent (SGD) optimization with a linearly decaying learning rate (given by $0.999(1 - s/2) + 0.001$ where s is the training step) and Nesterov momentum of 0.9. The models were trained for a maximum of 200 epochs but were stopped early if the loss on the validation dataset did not improve by at least 0.01 over 20 epochs. The number of epochs is limited in order to keep the ResNet training times comparable to SVMnet. The resulting accuracy and training time for each model was averaged over 5 repetitions of each experiment.

While the height and width of inputs can be adjusted for ResNet, the architecture always expects 3-channel RGB color images. Grayscale images were modified for use by ResNet by duplicating the pixel values into three equal channels. This approach was used in Section III-C and Section III-D. Before training and classification by ResNet, images were also passed through a preprocessing filter provided by the Keras library to prepare the data for ResNet models. All inputs were normalized by dividing by the mean and subtracting the variance before being used to train SVMnet. For RGB color inputs, the images were normalized per-channel.

All experiments and analysis presented in this section used the same hardware environment. SVMnet was parallelized across 16 cores of Intel Xeon Gold 6130 CPUs, and ResNet models were trained on an nVidia GeForce GTX 2080 GPU.

A. COIL-100 OBJECT RECOGNITION

Columbia Object Image Library (COIL-100) is a common dataset used for basic object recognition [33]. It contains RGB color images of 100 different objects, each photographed 72 times at 5 deg increments about the vertical axis. Background details were removed in all images and the objects are centered and enlarged to fill the frame. Some objects contained in this dataset include coffee mugs, small toy cars, and various fruits and vegetables.

The SVMnet in this experiment used one layer with a 25×25 window (giving each SVM 1875 input features) and a stride length of 7, followed by a weighted vote with dropout. The SVMnet and ResNet models were fitted with 100-500 training images in increments of 100, each controlled to have an equal number of samples for each object. A separate subset of 200 images was used as validation data for ResNet models.

Fig. 3 shows the results of this experiment. When fitted on the smallest training set, containing only one example per object, SVMnet correctly predicted labels for over 60% of the remaining images. With the same training set, ResNet-50 showed about the same accuracy and only

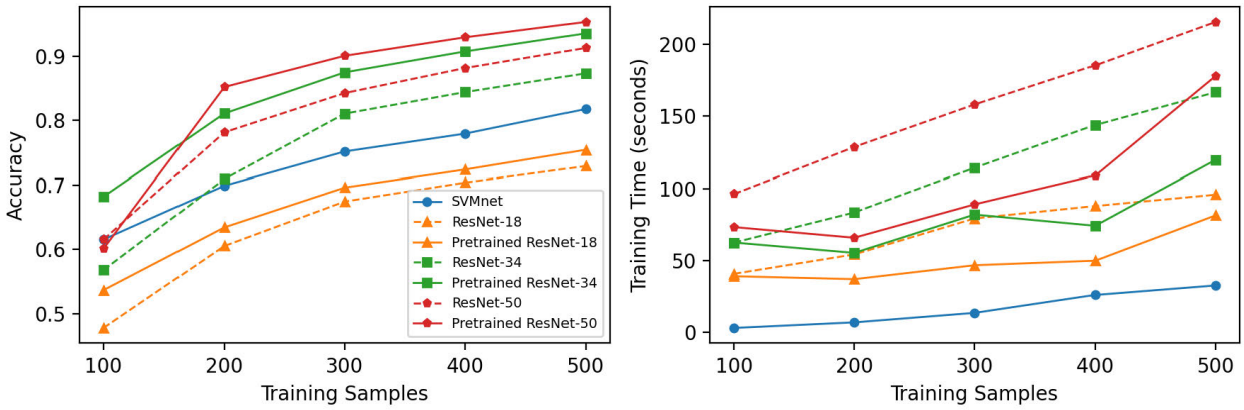

FIGURE 3. Test-set accuracy (left) and training time (right) of SVMnet and ResNet on COIL-100 images when fitted with different training set sizes.

pretrained ResNet-34 exceeded SVMnet; however, SVMnet was significantly faster to train in all cases.

B. IMAGENETTE

Imagenette is a fairly small, 10-class subset of the ImageNet dataset [34]. Several versions of this dataset exist; here we use version 2 of the 160 px dataset with noiseless labels. Many of these images are rectangular with their shortest side scaled to 160 px. In this experiment, we symmetrically zero-pad each image along its shorter axis to make it square, then downscale the images to have the same dimensions of 160 × 160 px.

The SVMnet used here contains one layer with a window size of 22 and stride length 7, followed by a weighted vote with no dropout. Imagenette is pre-divided into training and testing subsets containing 9,469 and 3,925 images, respectively. Models were trained using 20, 40, 80, 160, and 320 images from the provided training set and evaluated using the provided testing set. An additional 100 images were selected from the training set as validation data for the ResNet models.

Fig. 4 shows the results of this experiment. SVMnet achieved higher accuracy than all ResNet models for all training sets except the largest, where the ResNet-50 model pretrained with ImageNet weights improved drastically. The generally low accuracy of these models could be explained by the method used to conform each image to the same dimensions, which introduces a significant amount of empty space in many images. However, even under these conditions, SVMnet attained the highest accuracy in the least time for the smaller training sets.

C. COVID-19 RADIOGRAPHY

During the COVID-19 pandemic, machine learning techniques have been applied to various kinds of data to assist the medical community in making accurate diagnoses [35]–[37]. During the early stages of a disease outbreak, diagnostic data is expected to be limited or sparse, making it difficult to train most kinds of machine learning models. A type of model capable of learning from a small number of samples would be the most effective in this time frame.

Here we apply SVMnet to a database of chest x-ray images from healthy patients and patients diagnosed with COVID-19 [38], [39]. In this experiment, only the images labeled as "Normal" and "COVID" are used. Images were downscaled to 128 × 128 pixels (approx. 43% of the original size). An equal number of images were selected from each class, totaling 7232 samples. Models were fitted with 10, 20, 50, 100, and 200 training samples, with 50 separate images used as validation data for the ResNet models. The SVMnet uses two layers: the first with window size 19, stride 7, and class probability outputs; the second with window size 5 and stride 5, followed by an unweighted vote. During the architecture experiments described in Section III-G, the 2-layer SVMnet was shown to outperform the 1-layer models for this dataset.

Fig. 5 shows the results of this experiment. SVMnet was able to correctly label between 64% and 78% of unseen x-rays depending on the number of training samples, but most ResNet models failed to make significantly accurate predictions. Only the 18- and 34-layer ResNet models trained from scratch approached the accuracy of SVMnet. Additionally, SVMnet was several times faster to train.

D. ASTRONOMICAL IMAGE DATA

To test the performance of SVMnet on a current real-world image classification problem, a dataset of galaxy images from the Panoramic Survey Telescope and Rapid Response System (Pan-STARRS) was used. The dataset is made of galaxies separated into elliptical and spiral morphology. The galaxy images were taken from the catalog of Pan-STARRS galaxies classified by their morphological type [40].

An equal number of images were selected of each morphological type, totaling 26,732 samples. Each image is grayscale and has a dimension of 120 × 120 px. SVMnet and ResNet models were fitted with 10, 20, 40, 80, 160, and 320 training samples, with 200 separate images used as validation data for the ResNet models. The SVMnet uses one layer with a

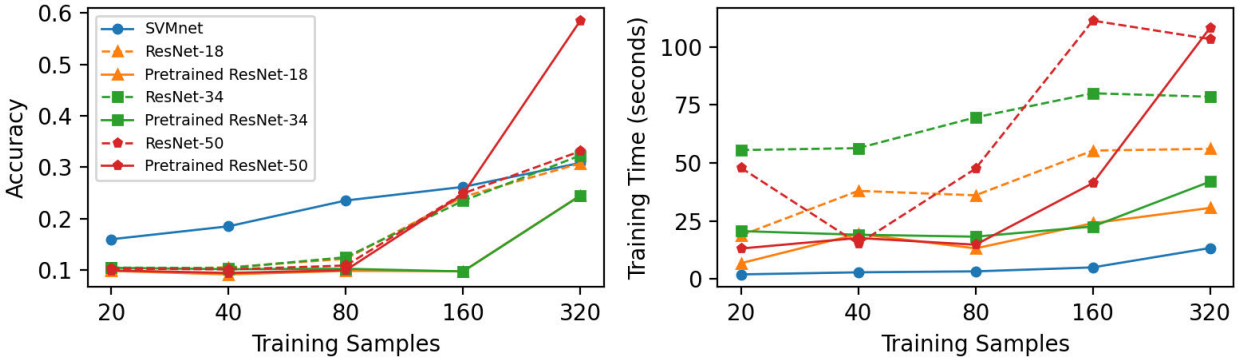

FIGURE 4. Test-set accuracy (left) and training time (right) of SVMnet and ResNet on Imagenette when fitted with different training set sizes.

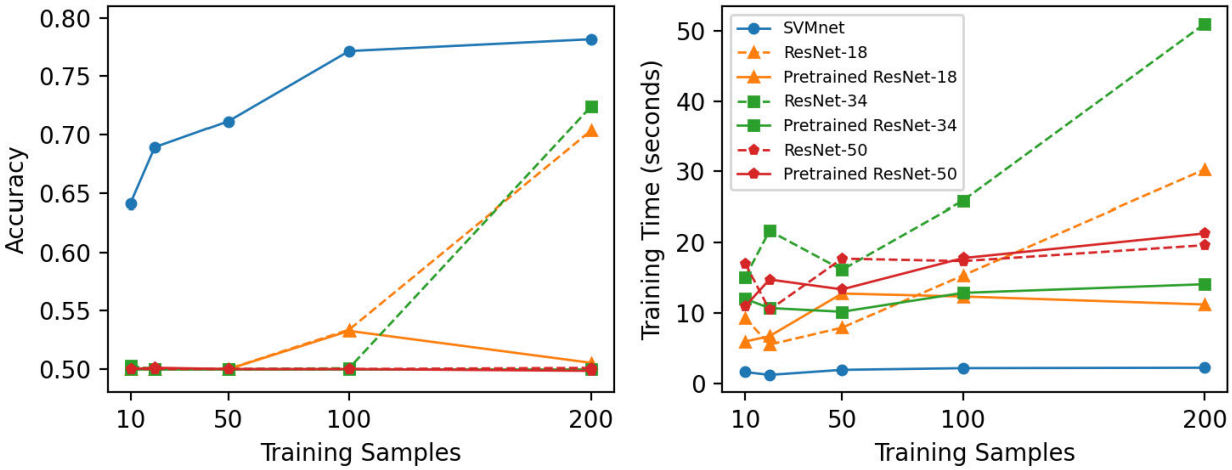

FIGURE 5. Test-set accuracy (left) and training time (left) of SVMnet and ResNet on COVID-19 chest x-ray images when fitted with different training set sizes. The accuracy of the ResNet models displays considerable overlap.

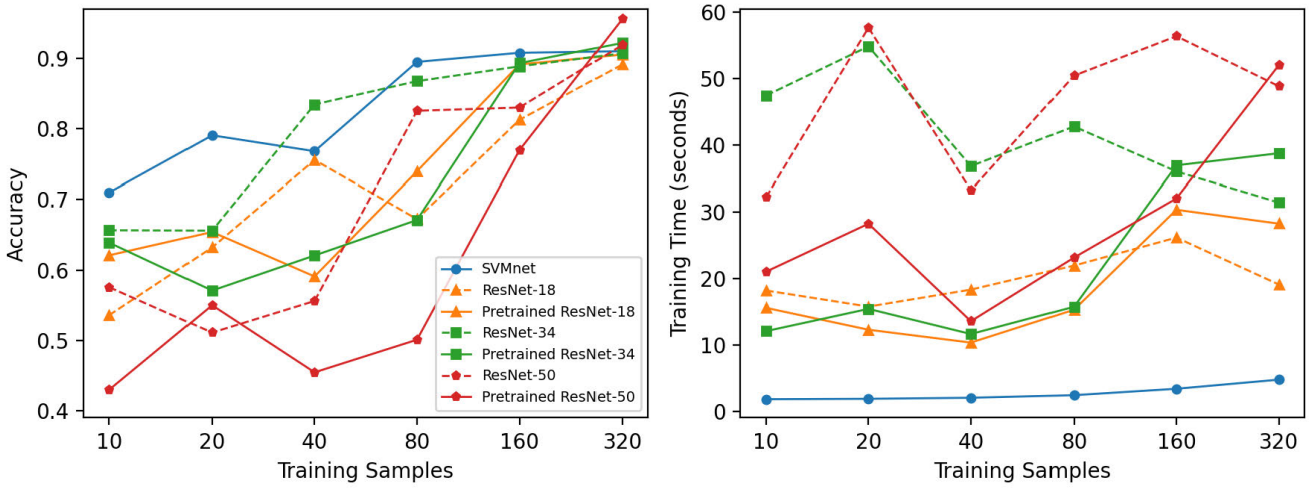

FIGURE 6. Test-set accuracy (left) and training time (right) of SVMnet and ResNet on Pan-STARRS galaxy images when fitted with different training set sizes.

window size of 22 and stride 5, followed by a weighted vote with dropout.

Fig. 6 shows the results of this experiment. As the graph shows, SVMnet outperformed almost every ResNet model when trained with a relatively small dataset. The models generally improve as the training set grows, with several ResNets slightly overtaking SVMnet with the largest training

TABLE 1. Comparison of the classification accuracy of WND-CHARM and SVMnet when trained on a small number of samples from four datasets.

COIL-100

	WND-CHARM	SVMnet
100	54%	62%
200	59%	70%
300	61%	75%
400	64%	78%

Imagenette

	WND-CHARM	SVMnet
20	11%	16%
40	13%	19%
80	16%	24%
160	18%	26%
320	21%	31%

COVID-19

	WND-CHARM	SVMnet
10	53%	64%
20	55%	69%
50	60%	71%
100	64%	77%
200	66%	78%

Pan-STARRS

	WND-CHARM	SVMnet
10	52%	71%
20	56%	79%
40	61%	77%
80	63%	90%
160	72%	91%
320	88%	91%

set. In all cases, SVMnet finished training many times faster than all ResNet models.

E. WND-CHARM

To test a "traditional" approach of using an SVM after extracting image features, we used the WND-CHARM open source feature set [41] combined with an SVM with linear kernel implemented through SVMLib. Table 1 compares the test set accuracy of WND-CHARM and SVMnet using the experimental datasets described earlier in this Section. WND-CHARM was trained on equal sized training subsets and consistently showed lower classification accuracy than SVMnet under the same conditions.

F. COMPUTATIONAL COMPLEXITY

The complexity of fitting an SVM is asymptotic and polynomial. For a training set containing n samples, the algorithm is dominated by either an n^2 term or an n^3 term based on the formulation of the problem [42]. Therefore, training a large number of SVMs can be a computationally demanding task, and can lead to substantial computational complexity during training.

The number of SVMs N in a layer receiving rectangular input with width I_x and height I_y is given by (4). The window size W (equivalent to the kernel size in other CNN literature), stride length S, and padding amount P in their respective dimensions follow from standard convolutional arithmetic. When using a square window on square input, the formula can be simplified to (5).

$$N = \left(\frac{I_x + 2P_x - W_x}{S_x} + 1\right) \cdot \left(\frac{I_y + 2P_y - W_y}{S_y} + 1\right) \quad (4)$$

$$N = \left(\frac{I + 2P - W}{S} + 1\right)^2 \quad (5)$$

Fitting a layer in SVMnet requires fitting N SVMs - a polynomial time operation. If the layer includes weights, then the SVMs must predict a class label for each input during the fit step, which scales linearly with the number of samples n. When using dropout as described in Section II-A, SVMnet performs an additional step during training that scales linearly with n. Thus, fitting SVMnet is dominated by the polynomial fit time of the SVMs. This relationship can be observed experimentally in Fig. 8.

CNNs can theoretically be trained infinitely, but there is a definitive point at which the SVMs within SVMnet converge. This places a soft upper bound on the training time of SVMnet based on the tolerance parameter of the SVMs. Additionally, a firm upper bound may be placed on the number of iterations of the SVM algorithm, allowing for a shortened training time at the expense of some accuracy.

SVMnet trains multiple SVMs simultaneously using process-based parallelism and shared memory, greatly increasing its speed on typical multicore computers with minimal overhead. While this allows SVMnet to run quite easily on relatively inexpensive systems, the potential performance gain from extra hardware is minimal compared to the extreme optimization of CNNs for GPU devices.

While the training of SVMnet is slower than CNNs when the size of the training set becomes relatively large, SVMnet is designed for situations in which the size of the training set is small. Therefore, the computational complexity of the training is not expected to introduce a major obstacle in many real-world cases where the size of the training set is limited, and the time required for training does not necessarily explode to an unmanageable response time in the situations where SVMnet is most effective.

1) INFERENCE TIME OF IMAGE CLASSIFICATION

Predicting a single class label of an image using SVMnet typically requires a large number of individual SVMs to predict a label followed by a vote tally. Despite its affinity for parallelization, this process is expected to take longer than the highly optimized matrix operations of a CNN. Table 2 compares the inference time of SVMnet and ResNet on images in the COIL-100 dataset.

TABLE 2. Comparison of the response time (in seconds) of SVMnet and ResNet to predict class labels for 1, 10, 100, and 1000 samples of the COIL-100 dataset.

	1	10	100	1000
SVMnet	2.36	2.66	3.81	24.2
ResNet-18	0.054	0.056	0.082	0.296
ResNet-34	0.060	0.060	0.098	0.410
ResNet-50	0.061	0.064	0.106	0.515

The comparison shows that SVMnet is significantly slower than ResNet for classifying samples, but the speed of classification is still practical for many real-world systems. The parallelization of SVMnet greatly reduces the time needed to make predictions, but the overhead of shared memory operations is significant in the case of few samples.

G. ARCHITECTURE COMPARISON

As with CNNs, SVMnet can be configured into a variety of architectures which are expected to differ in performance depending on the classification task. Due to the high number

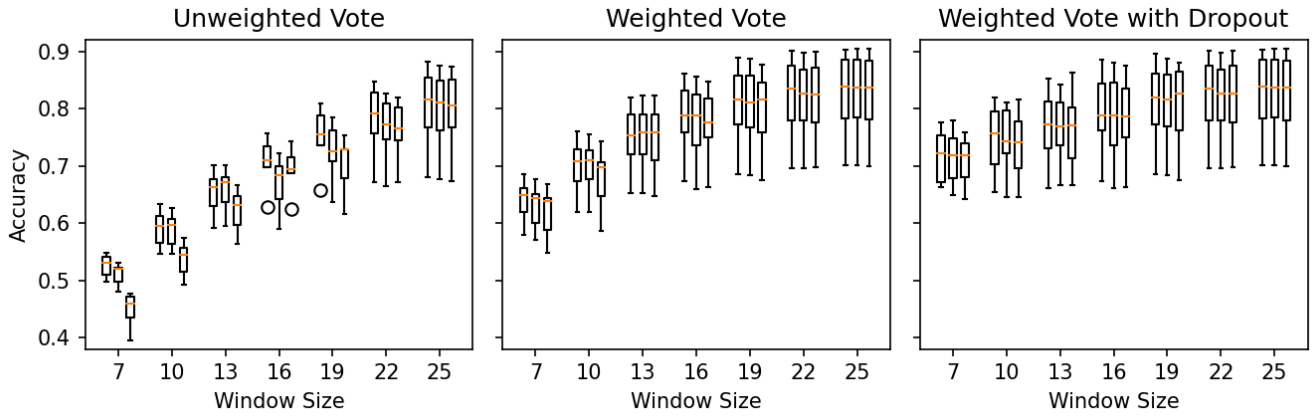

FIGURE 7. Prediction accuracy of one-layer SVMnet architectures fitted to COIL-100. Each group of three box plots represents the same window size with stride length 3, 5, and 7, respectively. Each box plot shows the distribution in model accuracy when using five training sets of 200-1000 examples.

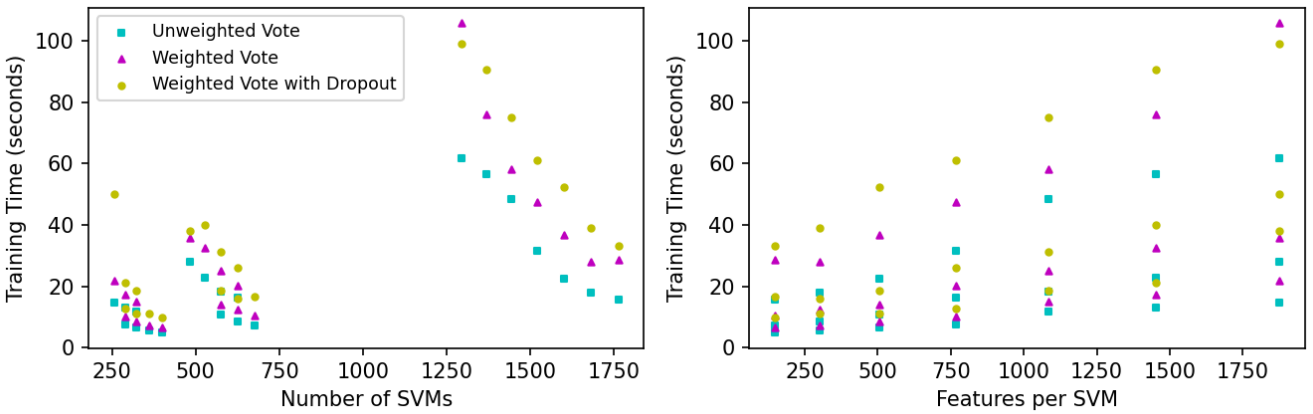

FIGURE 8. Training times for one-layer SVMnet architectures fitted to COIL-100.

of possible models, determining which is the most effective for a single task is non-trivial. In this section we show how a variety of SVMnet configurations were tested on the COIL-100 dataset to inform the choice of model used in Section III-A. Similar methods were used to select the models for other datasets. SVMnet models with multiple layers were tested in the same manner.

Fig. 7 shows how the performance of a one-layer SVMnet changes with the window size, stride length, voting method, and number of training samples when fitted to COIL-100. Prediction accuracy improves in all cases as the window size increases but with diminishing returns. Increasing the stride length tends to lower accuracy when the window is small but incurs little to no penalty when the window is large. When the vote of an SVM is weighted, model accuracy improves in all cases compared to an unweighted vote; performance increases further when using dropout as described in Section II-A. This effect is more significant when the window size is small.

Fig. 8 shows how the time required to fit SVMnet on COIL-100 changes with the number of SVMs (see Equation 5) and the number of features for each SVM (in this case equal to $3W^2$). Since increasing the stride length significantly reduces the number of SVMs in the model, an SVMnet with large windows can still be trained quickly with only a minor increase in stride without sacrificing accuracy. Each SVMnet in this experiment was trained in parallel using 16 CPUs.

IV. CONCLUSION

Deep convolutional neural networks provide excellent performance in automatic classification of image data while eliminating the need to develop and tailor algorithms for specific image classification problems. With the availability of open source libraries, DCNNs have become the de facto first solution to image classification.

Here we explore one of the primary weaknesses of DCNNs, which is the need of a relatively high number of labeled "ground truth" samples for effective training of the network. While in the computer science literature DCNNs are often tested on relatively large datasets such as MNIST or ImageNet, in many real-world problems a very large number of clean labeled samples that can be used for training is not available.

Medical datasets such as those prepared for the purpose of image-based diagnostics are difficult to prepare due to the long time required to assign a sample with a correct label [5], consequently leading to a high cost. Additionally, acquiring a radiograph can also require substantial resources, as medical image acquisition systems such as Magnetic Resonance Imaging (MRI) require expensive instrumentation and staff. Additionally, the consent of the patient is required for the preparation of each sample. These limitations make large datasets of biomedical images substantially more expensive and more difficult to prepare.

In many other cases labeled training samples are not available. For instance, when analyzing archaeological artifacts, the number of training samples are limited by the number of available artifacts, which is often a hard limit that cannot be easily changed. A typical size of such datasets is normally several hundred samples [43]. Using computer vision to analyze art [44] is limited by the number of paintings each artists created, which can be a firm limit, especially when the painter is no longer alive. These are obviously just a few examples out of many possible real-world situations in which the number of labeled samples is inherently small.

SVMnet aims at providing an effective solution for the numerous real-world situations in which the number of labeled image samples that can be used for training is limited. SVMnet utilizes the ability of an SVM to learn from a smaller number of samples compared to other machine learning approaches. The flexible structure of SVMnet allows it to learn directly from the pixel values, and to utilize different layers that correspond to the convolutional and fully connected layers in ''conventional'' deep neural networks.

Like DCNNs, SVMnet does not require the design of specific algorithms for a particular image classification problem. Therefore, SVMnet can be used for a variety of image data, as also demonstrated in Section III. One of its primary uses can be the biomedical domain, where the acquisition and annotation of images is expensive and time-consuming, and therefore biomedical datasets are very often much smaller than image datasets used in other tasks such as object recognition.

The proposed approach is structured as a network to take advantage of the stronger signal from neighboring pixels, similar to the core idea in the basis of CNNs. SVMs are known for their ability to learn quickly from relatively few training samples. By training many SVMs on small pixel regions across an image, this quick learning can be leveraged to extract much information from small sets of images in less time than it would take to fully train a deep neural network.

Complexity analysis shows that the training time for SVMnet scales more quickly with the number of input samples than DCNNs, suggesting that SVMnet might take substantial computational resources when trained using large datasets. However, SVMnet is designed for situations in which the labeled training set is relatively small. As shown in our experiments, the training time might not be a practical obstacle in many real-world situations in which SVMnet can be used. While computing is an available resource, and training SVMnet with a few hundred training samples scales within reasonable response time, annotated clean or rare training samples might in many cases be much more difficult to obtain.

The underlying structure used to create SVMnet is very flexible, allowing other kinds of machine learning algorithms to be used rather than solely SVMs. Constructing the layers with classifiers such as random forests or logistic regression may result in better performance for some datasets. These layers can be mixed in the same model as well, i.e. using one layer of SVMs followed by a layer of random forests. These possibilities present a promising avenue for future related work.

SVMnet is not designed to become a general solution that can outperform deep convolutional neural networks such as ResNet-50. But experimental results show that it is an effective solution for cases in which the number of labeled training samples is small. Since such cases are not rare, SVMnet can complement conventional deep neural networks by providing image classification in the cases where not many labeled training samples are available.

REFERENCES

[1] A. Khan, A. Sohail, U. Zahoora, and A. S. Qureshi, ''A survey of the recent architectures of deep convolutional neural networks,'' *Artif. Intell. Rev.*, vol. 53, no. 8, pp. 5455–5516, 2020.

[2] K. He, X. Zhang, S. Ren, and J. Sun, ''Deep residual learning for image recognition,'' in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[3] J. A. Nichols, H. W. Herbert Chan, and M. A. B. Baker, ''Machine learning: Applications of artificial intelligence to imaging and diagnosis,'' *Biophys. Rev.*, vol. 11, no. 1, pp. 111–118, Feb. 2019.

[4] T. T. Tang, J. A. Zawaski, K. N. Francis, A. A. Qutub, and M. W. Gaber, ''Image-based classification of tumor type and growth rate using machine learning: A preclinical study,'' *Sci. Rep.*, vol. 9, no. 1, Dec. 2019, Art. no. 12529.

[5] C. Martin-Isla, V. M. Campello, C. Izquierdo, Z. Raisi-Estabragh, B. Baeßler, S. E. Petersen, and K. Lekadir, ''Image-based cardiac diagnosis with machine learning: A review,'' *Frontiers Cardiovascular Med.*, vol. 7, p. 1, Jan. 2020.

[6] A. Zhang, S. Li, Y. Cui, W. Yang, R. Dong, and J. Hu, ''Limited data rolling bearing fault diagnosis with few-shot learning,'' *IEEE Access*, vol. 7, pp. 110895–110904, 2019.

[7] M. Poostchi, K. Silamut, R. J. Maude, S. Jaeger, and G. Thoma, ''Image analysis and machine learning for detecting malaria,'' *Transl. Res.*, vol. 194, pp. 36–55, Apr. 2018.

[8] N. V. Orlov, A. T. Weeraratna, S. M. Hewitt, C. E. Coletta, J. D. Delaney, D. Mark Eckley, L. Shamir, and I. G. Goldberg, ''Automatic detection of melanoma progression by histological analysis of secondary sites,'' *Cytometry A*, vol. 81A, no. 5, pp. 364–373, May 2012.

[9] J. Ker, Y. Bai, H. Y. Lee, J. Rao, and L. Wang, ''Automated brain histology classification using machine learning,'' *J. Clin. Neurosci.*, vol. 66, pp. 239–245, Aug. 2019.

[10] A. Gertych, N. Ing, Z. Ma, T. J. Fuchs, S. Salman, S. Mohanty, S. Bhele, A. Velásquez-Vacca, M. B. Amin, and B. S. Knudsen, ''Machine learning approaches to analyze histological images of tissues from radical prostatectomies,'' *Comput. Med. Imag. Graph.*, vol. 46, pp. 197–208, Dec. 2015.

[11] J. W. Vaughan, ''Making better use of the crowd: How crowdsourcing can advance machine learning research,'' *J. Mach. Learn. Res.*, vol. 18, no. 1, pp. 7026–7071, 2017.

[12] V. S. Sheng and J. Zhang, ''Machine learning with crowdsourcing: A brief summary of the past research and future directions,'' in *Proc. AAAI Conf. Artif. Intell.*, vol. 33, 2019, pp. 9837–9843.

[13] N. Zhou, Z. D. Siegel, S. Zarecor, N. Lee, D. A. Campbell, C. M. Andorf, D. Nettleton, C. J. Lawrence-Dill, B. Ganapathysubramanian, J. W. Kelly, and I. Friedberg, "Crowdsourcing image analysis for plant phenomics to generate ground truth data for machine learning," *PLOS Comput. Biol.*, vol. 14, no. 7, Jul. 2018, Art. no. e1006337.
[14] L. Shamir, C. Yerby, R. Simpson, A. M. von Benda-Beckmann, P. Tyack, F. Samarra, P. Miller, and J. Wallin, "Classification of large acoustic datasets using machine learning and crowdsourcing: Application to whale calls," *J. Acoust. Soc. Amer.*, vol. 135, no. 2, pp. 953–962, Feb. 2014.
[15] M. Lease, "On quality control and machine learning in crowdsourcing," in *Proc. 25th AAAI Conf. Artif. Intell.*, 2011, pp. 97–102.
[16] L. Fei-Fei, R. Fergus, and P. Perona, "One-shot learning of object categories," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 4, pp. 594–611, Apr. 2006.
[17] L. Wolf, T. Hassner, and Y. Taigman, "The one-shot similarity kernel," in *Proc. IEEE 12th Int. Conf. Comput. Vis.*, Sep. 2009, pp. 897–902.
[18] G. Koch, R. Zemel, and R. Salakhutdinov, "Siamese neural networks for one-shot image recognition," in *Proc. ICML deep Learn. Workshop*, vol. 2, Lille, France, 2015, pp. 1–30.
[19] O. Vinyals, C. Blundell, T. Lillicrap, K. Kavukcuoglu, and D. Wierstra, "Matching networks for one shot learning," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 29, 2016, pp. 3630–3638.
[20] G. Bender, P.-J. Kindermans, B. Zoph, V. Vasudevan, and Q. Le, "Understanding and simplifying one-shot architecture search," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 550–559.
[21] X. Tang, F. Meng, X. Zhang, and Y. Cheung, "Hyperspectral image classification based on 3-D octave convolution with spatial–spectral attention network," *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 3, pp. 2430–2447, Mar. 2021.
[22] J. Bai, B. Ding, Z. Xiao, L. Jiao, H. Chen, and A. C. Regan, "Hyperspectral image classification based on deep attention graph convolutional network," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, pp. 1–16, 2022.
[23] C. Cortes and V. Vapnik, "Support-vector networks," *Mach. Learn.*, vol. 20, no. 3, pp. 273–297, Jul. 1995.
[24] W. Chen, S.-H. Hsu, and H.-P. Shen, "Application of SVM and ANN for intrusion detection," *Comput. Oper. Res.*, vol. 32, no. 10, pp. 2617–2634, Oct. 2005.
[25] S. Mukkamala, G. Janoski, and A. Sung, "Intrusion detection using neural networks and support vector machines," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, vol. 2, May 2002, pp. 1702–1707.
[26] L. Zhang, W. Zhou, and L. Jiao, "Wavelet support vector machine," *IEEE Trans. Syst., Man, Cybern. B. Cybern.*, vol. 34, no. 1, pp. 34–39, Oct. 2004.
[27] K.-S. Shin, T. S. Lee, and H.-J. Kim, "An application of support vector machines in bankruptcy prediction model," *Expert Syst. Appl.*, vol. 28, no. 1, pp. 127–135, Jan. 2005.
[28] E. Byvatov, U. Fechner, J. Sadowski, and G. Schneider, "Comparison of support vector machine and artificial neural network systems for drug/nondrug classification," *J. Chem. Inf. Comput. Sci.*, vol. 43, no. 6, pp. 1882–1889, Nov. 2003.
[29] L. Jiao, L. Bo, and L. Wang, "Fast sparse approximation for least squares support vector machine," *IEEE Trans. Neural Netw.*, vol. 18, no. 3, pp. 685–697, May 2007.
[30] J. S. Paiva, J. Cardoso, and T. Pereira, "Supervised learning methods for pathological arterial pulse wave differentiation: A SVM and neural networks approach," *Int. J. Med. Informat.*, vol. 109, pp. 30–38, Jan. 2018.
[31] V. Dumoulin and F. Visin, "A guide to convolution arithmetic for deep learning," 2018, *arXiv:1603.07285*.
[32] C. C. Chang and C. J. Lin, "LIBSVM: A library for support vector machines," *ACM Trans. Intell. Syst. Technol.*, vol. 2, no. 3, pp. 1–27, 2011.
[33] S. A. Nene *et al.*, "Columbia object image library (COIL-100)," Columbia Univ., New York, NY, USA, Tech. Rep. CUCS-006-96, 1996.
[34] J. Howard and S. Gugger, "Fastai: A layered API for deep learning," *Information*, vol. 11, no. 2, p. 108, 2020.
[35] E. El-Din Hemdan, M. A. Shouman, and M. Esmail Karar, "COVIDX-net: A framework of deep learning classifiers to diagnose COVID-19 in X-ray images," 2020, *arXiv:2003.11055*.
[36] C. Gangloff, S. Rafi, G. Bouzillé, L. Soulat, and M. Cuggia, "Machine learning is the key to diagnose COVID-19: A proof-of-concept study," *Sci. Rep.*, vol. 11, no. 1, Mar. 2021, Art. no. 7166.
[37] W. T. Li, J. Ma, N. Shende, and G. Castaneda, "Using machine learning of clinical data to diagnose COVID-19: A systematic review and meta-analysis," *BMC Med. Informat. Decis. Making*, vol. 20, no. 1, p. 247, Sep. 2020.
[38] E. H. Muhammad Chowdhury, T. Rahman, A. Khandakar, R. Mazhar, M. A. Kadir, Z. B. Mahbub, K. R. Islam, M. S. Khan, A. Iqbal, N. A. Emadi, M. B. I. Reaz, and M. T. Islam, "Can ai help in screening viral and COVID-19 pneumonia? *IEEE Access*, vol. 8, pp. 132665–132676, 2020.
[39] T. Rahman, A. Khandakar, Y. Qiblawey, A. Tahir, S. Kiranyaz, S. B. A. Kashem, M. T. Islam, S. Al Maadeed, S. M. Zughaier, M. S. Khan, and M. E. H. Chowdhury, "Exploring the effect of image enhancement techniques on COVID-19 detection using chest X-ray images," *Comput. Biol. Med.*, vol. 132, May 2021, Art. no. 104319.
[40] H. Goddard and L. Shamir, "A catalog of broad morphology of pan-STARRS galaxies based on deep learning," *Astrophys. J. Suppl. Ser.*, vol. 251, no. 2, p. 28, Dec. 2020.
[41] L. Shamir, N. Orlov, D. M. Eckley, T. Macura, J. Johnston, and I. G. Goldberg, "Wndchrm—An open source utility for biological image analysis," *Source Code Biol. Med.*, vol. 3, no. 1, pp. 1–13, Dec. 2008.
[42] L. Bottou and C.-J. Lin, "Support vector machine solvers," *Large Scale Kernel Mach.*, vol. 3, no. 1, pp. 301–320, 2007.
[43] M. P. Pavan Kumar, B. Poornima, H. S. Nagendraswamy, C. Manjunath, B. E. Rangaswamy, M. Varsha, and H. P. Vinutha, "Image abstraction framework as a pre-processing technique for accurate classification of archaeological monuments using machine learning approaches," *Social Netw. Comput. Sci.*, vol. 3, no. 1, pp. 1–30, Jan. 2022.
[44] F. S. Khan, S. Beigpour, J. van de Weijer, and M. Felsberg, "Painting-91: A large scale database for computational painting categorization," *Mach. Vis. Appl.*, vol. 25, no. 6, pp. 1385–1397, Aug. 2014.

HUNTER GODDARD received the B.S. and M.S. degrees in computer science from Kansas State University, Manhattan, KS, USA, in 2017 and 2021, respectively, where he is currently pursuing the Ph.D. degree in computer science. He has published one scientific paper in the *Astrophysical Journal Supplement Series* (ApJS) relating to research done for the M.S. degree. His research interests include machine learning algorithms and their applications in various fields including astronomy, bioinformatics, and medicine.

LIOR SHAMIR is currently an Associate Professor of computer science at Kansas State University, Manhattan, KS, USA. He is the author of one book and more than 100 scientific papers. His research interests include data science and the application of machine learning and artificial intelligence for the purpose of discovery from data. He is a member of the Midwest Big Data Hub (MBDH), the Scientific Advisory Board of the Astrophysics Source Code Library (ASCL), and the Informatics and Statistical Science Collaboration (ISSC) of the Vera Rubin Observatory.