**Performance and Resource Trade-Off Analysis of Neural Networks and Support Vector Machines on Pneumonia Detection**

Mary Klimasewiski, Leonardo Pinheiro, Sedat Touray

**1. Introduction**

In this study, we aim to explore the trade-offs between traditional machine learning models, such as Support Vector Machines (SVMs), and more advanced deep learning models, including Feed-Forward Neural Networks (FFNNs), in the context of pneumonia detection from chest X-rays.

While Convolution Neural Networks (CNNs) are widely regarded as the benchmark for image classification tasks due to their proven performance, our primary focus is on comparing SVMs and FFNNs directly. CNNs will serve as a reference point to understand the upper bound of model performance, but we are specifically interested in evaluating the trade-offs between SVM and FFNN, both of which offer different strengths and computational costs.

Support Vector Machines have long been used for classification tasks these models are fitted by determining hyperplanes that best separate different classes in a high-dimensional feature space, such as distinguishing between healthy and pneumonia-affected lung tissue in medical images. SVMs are particularly effective for smaller datasets and offer lower computational cost, but they can be sensitive to hyperparameter choices and may struggle with high-dimensional, complex data like raw images. Moreover, SVMs typically require manual feature engineering, which can be time-consuming and may limit their ability to automatically capture intricate patterns in the data.

On the other hand, neural networks, particularly FFNNs, are capable of learning hierarchical representations directly from raw data, such as pixel values in images, without the need for manual feature extraction. FFNNs are a more recent advancement in machine learning that can model complex, non-linear relationships within the data. While FFNNs can handle high-dimensional data and automatically learn patterns from raw images, they still require significant computational resources, although typically less than CNNs.

CNNs, in particular, are widely used as the benchmark in image classification tasks due to their ability to capture spatial hierarchies through convolutional layers. This makes them highly effective for pneumonia detection, where subtle patterns in X-ray images are key to accurate classification. Given their proven success in similar tasks, CNNs will serve as a benchmark model in this study. However, while CNNs offer superior performance, they come with significant trade-offs, particularly in terms of computational costs. Training CNNs requires large datasets, substantial memory, and powerful hardware such as Graphics Processing Units (GPUs), which can limit their practical deployment in resource-constrained environments.

Thus, this study will evaluate the trade-offs between traditional machine learning models (SVMs) and more advanced deep learning models (FFNNs) for pneumonia detection, using CNNs as a benchmark. We will assess these models using classification performance metrics (accuracy, precision, recall, F1-score), as well as computational efficiency (runtime, memory usage, and hardware requirements).

**2. Methodology**

**The RSNA Pneumonia Detection Dataset**

The RSNA Pneumonia Detection Challenge 2018, hosted by the Radiological Society of North America (RSNA) in collaboration with the National Institutes of Health (NIH) and Stanford University, provides a comprehensive dataset for pneumonia detection from chest X-ray images.

The dataset was sourced from the National Institutes of Health (NIH)'s publicly available Chest X-ray 8 (CXR8) collection. This dataset comprises over 112,000 frontal-view chest radiographs, each labeled with 14 different disease labels, including pneumonia.

For the challenge, a subset of 30,000 images was selected, consisting of 16,248 posteroanterior views and 13,752 anteroposterior views. These images were converted from Portable Network Graphics (PNG) format to Digital Imaging and Communications in Medicine (DICOM) format to standardize the dataset for medical imaging applications.

The dataset was meticulously annotated by multiple expert reviewers, including specialists from the Society of Thoracic Radiology. These professionals identified abnormal areas in the lung images and assessed the probability of pneumonia, providing the ground truth for participants to train and evaluate their algorithms.

For our study we selected a smaller subset The RSNA dataset consists of 5,840 frontal-view chest X-ray The dataset includes 1,575 images without any signs of pneumonia.

and 4,265 images indicating the presence of pneumonia, with varying levels of severity.

This comprehensive dataset has been instrumental in advancing research in automated pneumonia detection, offering a valuable resource for developing and evaluating machine learning models in the medical imaging domain.

**Dataset Description**

The RSNA dataset consists of 5,840 frontal-view chest X-ray images sourced from patients with varying levels of pneumonia severity. The dataset includes:

* Normal Class (1,575 images): Chest X-rays from healthy individuals without any signs of pneumonia.
* Pneumonia Class (4,265 images): Chest X-rays indicating the presence of pneumonia, with varying levels of severity.

Each image is annotated with labels from radiologists to indicate the presence or absence of pneumonia. The images are in DICOM format (Digital Imaging and Communications in Medicine), commonly used in medical imaging.

A collage of x-ray images of a person's chest

AI-generated content may be incorrect.

Figure 1 Sample Images from Dataset

**Preprocessing Steps**

The raw chest X-ray images require several preprocessing steps before being fed into machine learning models to ensure optimal performance and compatibility. DICOM images were processed using the pydicom library, which allows easy manipulation of DICOM files. These images were converted into NumPy arrays for further processing. All images were resized to a uniform size of 224x224 pixels to maintain consistency in dimensions, making them suitable for input into neural networks. This resizing was done using cv2 or PIL (Python Imaging Library) to avoid potential distortions and loss of crucial image features.

For the SVM model, we applied Principal Component Analysis (PCA) to reduce the dimensionality of the data. After flattening the images into vectors, 50 principal components were retained, explaining most of the variance in the data. This helped reduce the computational cost for SVM without significant loss of performance. For SVM and FFNN models, feature scaling was applied. The pixel values were standardized using zero-mean and unit-variance transformation, which normalizes the feature distribution to improve model convergence.

The dataset was split into a 70-30% train-test set, with 70% of the images used for training and 30% reserved for testing. This split ensures that the model is evaluated on unseen data and helps prevent overfitting. Furthermore, to handle class imbalance (as pneumonia images are more abundant), a stratified shuffle split was employed to maintain the same proportion of classes in both training and validation sets.

t-Distributed Stochastic Neighbor Embedding (t-SNE) was used for visualizing the high-dimensional feature space in 2D and 3D. This helped assess how well the models were separating the two classes (normal and pneumonia) after training, especially for the FFNN and CNN models.

Completing these preprocessing steps prepared the data for the application of our machine learning models, ensuring that they could learn effectively from the images while maintaining generalizability and performance.

**Machine Learning Models and Theoretical Foundations**

**Support Vector Machine (SVM)**

Developed in the 1990s by Vapnik and Cortes, SVMs are among the most widely used classification algorithms due to their strong theoretical foundation in statistical learning theory (Cortes & Vapnik, 1995).

SVMs classify data by finding an optimal hyperplane that maximizes the margin between different classes. The decision function is given by

Where are Lagrange multipliers, are class labels, is a kernel function, is the bias term. For this study, we used a linear kernel with C = 1.0, achieving an accuracy of 77.5%.

<image: Visualization of SVM decision boundary with support vectors>

**Feed-Forward Neural Network (FFNN)**

Inspired by biological neurons, FFNNs were introduced in the 1940s by McCulloch and Pitts and later formalized by Rosenblatt’s perceptron model (1958). However, it was not until the development of backpropagation (Rumelhart, Hinton, & Williams, 1986) that neural networks became practical for real-world applications.

A fully connected feed-forward network processes inputs through multiple layers of neurons, where each neuron computes

where are weight matrices, are bias terms, and is an activation function (ReLU in our case).

We used the following FFNN architecture, an input Layer taking in flattened pixel values from the X-ray images, two hidden layers with 64 and 32 neurons, using ReLU activation and an output layer with softmax activation for binary classification

A black background with a black square

AI-generated content may be incorrect.

**Convolutional Neural Network (CNN)**

Introduced by LeCun et al. (1998), CNNs revolutionized computer vision by learning hierarchical feature representations directly from images. Modern CNN architectures such as AlexNet, VGG, and ResNet have demonstrated state-of-the-art performance in image classification.

CNNs use convolutional layers to detect spatial patterns in images. The core operation is:

Where is the filter matrix, is the input feature map, and is the bias term.

We used the following CNN Architecture. One convolutional Layer with 32 filters, 3×3 kernel, ReLU activation, one pooling Layer with 2×2 max pooling, a fully connected Layer with128 neurons with dropout regularization and one output layer for softmax activation

<image: CNN feature extraction process with convolutional layers>

**Model Performance and Computational Trade-Offs**

**Model Performance Results**

**SVM**

**Model Performance**

**Accuracy: 0.7746689982513115**

**Performance Report:**

Test Accuracy: 0.7746689982513115

precision recall f1-score support

0 0.77 1.00 0.87 3101

1 0.00 0.00 0.00 902

**Computational Efficiency**

Runtime: 9.95 seconds

CPU Usage: 3.10%

Memory Used: 346.75 MB

GPU Memory Used: N/A MB

**FFNN**

Runtime: 14.77 seconds

CPU Usage: 14.80%

Memory Used: 5854.625MB

A graph with a line

AI-generated content may be incorrect.

**Conclusion**

This study indicates, as expected, that CNNs outperform both SVMs and FFNNs in terms of accuracy for pneumonia detection. CNNs' ability to automatically extract hierarchical features from images gives them a clear advantage in image classification tasks. However, this superior performance comes at a cost: CNNs require significantly more computational resources, including high memory usage, longer training times, and often the need for specialized hardware like GPUs.

For environments with limited hardware resources, such as mobile devices, small clinics, or real-time applications where computational efficiency is crucial, deploying CNNs may not be practical. In such cases, SVMs or smaller FFNNs present viable alternatives. While they may exhibit a slight trade-off in accuracy, they are considerably more lightweight, requiring less memory and processing power. SVMs, in particular, perform well on smaller datasets and structured features, while FFNNs can still leverage deep learning's advantages without the full computational burden of CNNs.

Ultimately, the choice between these models depends on the specific constraints of the deployment setting. If maximizing accuracy is the primary goal and computational resources are available, CNNs remain the best option. However, if real-time performance, lower memory consumption, or deployment on resource-limited hardware is a priority, SVMs or FFNNs provide practical alternatives that balance performance and feasibility.

**References**

Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine Learning, 20*(3), 273-297.

Scholkopf, B., & Smola, A. J. (2002). *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond.* MIT Press.

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. *Nature, 323*, 533-536.

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning.* MIT Press.

LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE, 86*(11), 2278-2324.