Communications on Applied Nonlinear Analysis
ISSN: 1074-133X
Vol 32 No. 2 (2025)

# VMCNN: Integrating Vedic Mathematics for Enhanced Convolutional Neural Network Performance in Image Classification Tasks

**Geetanjali Kale (Rao)** [1*] **Dr. Seema Raut**[2**]
[1,2]G H Raisoni University, Amravati, India

***Article History:***

***Received:*** *19-07-2024*

***Revised:*** *19-09-2024*

***Accepted:*** *27-09-2024*

**Abstract:**

In the rapidly advancing field of deep learning, Convolutional Neural Networks (CNNs) have emerged as a cornerstone for various image classification tasks. However, the existing CNN architectures and optimization methods often grapple with trade-offs between computational efficiency, accuracy, and generalizability. This work addresses these limitations by integrating principles of Vedic Mathematics, an ancient Indian mathematical system known for its efficiency and simplicity, into CNN optimization. Traditional optimization techniques often fall short in balancing the computational load with the accuracy of the model, particularly in diverse and complex datasets. To overcome these challenges, we propose a novel model that incorporates specific Vedic Mathematical Sutras, namely Urdhva-Tiryakbhyam, Anurupyena, Nikhilam Navatashcaramam Dashatah, Shunyam Saamyasamuccaye, Paravartya Yojayet, Sankalana-vyavakalanabhyam, Ekadhikina Purvena, and Gunitasamuchyah, for optimizing internal operations of CNNs. These Sutras were meticulously selected for their potential to simplify computational processes, enhance parallel processing capabilities, and optimize training algorithms. The application of these sutras has led to a remarkable improvement in the performance of CNNs across various datasets, including ImageNet, CIFAR, ChestXRay8, and Architectural Heritage Datasets. The optimized CNN models demonstrated a significant enhancement in classification accuracy (9.5% increase), precision (8.3% increase), recall (8.5% increase), area under the curve (AUC) (4.9% increase), MAE (5.5% improvement), and reduction in delay (6.5% decrease) compared to existing methods. The integration of Vedic Mathematics into CNN optimization not only paves the way for more efficient and accurate image classification models but also opens new avenues for interdisciplinary research, blending ancient mathematical wisdom with contemporary artificial intelligence techniques. This advancement has profound implications for various applications, including medical imaging, autonomous vehicles, and heritage conservation, thereby contributing significantly to the field of AI and computational efficiency.

**Keywords:** Convolutional Neural Networks, Vedic Mathematics, Image Classification, Computational Efficiency, Algorithm Optimization

## 1. Introduction

In the domain of deep learning, Convolutional Neural Networks (CNNs) have been at the forefront of breakthroughs in image classification. Their ability to extract and learn features from images has made them indispensable in a variety of applications, ranging from medical diagnostics to autonomous vehicle navigation. However, as the demand for higher accuracy and computational efficiency grows, traditional CNN architectures and optimization methods increasingly face challenges in meeting these demands, particularly when processing diverse and complex datasets.

The quest for optimizing CNNs has led researchers to explore various avenues, from architectural innovations to advanced training algorithms. Yet, these efforts often encounter limitations in balancing computational load, model accuracy, and generalizability. This paper addresses these challenges by turning to a rather unconventional source: Vedic Mathematics. This ancient Indian mathematical system is renowned for its simplified and efficient computational techniques, which have the potential to revolutionize the way CNNs are optimized.

Vedic Mathematics, with its array of Sutras (formulas or principles), offers a unique approach to simplifying complex calculations. In this study, we harness this potential to enhance the performance of CNNs. We meticulously integrate specific Vedic Sutras such as Urdhva-Tiryakbhyam for efficient matrix multiplication, Anurupyena for scaling operations, and Nikhilam Navatashcaramam Dashatah for optimized subtraction processes, among others. These Sutras were chosen for their relevance to the computational processes integral to CNN operations, such as convolution, pooling, and backpropagation.

The application of these principles to CNN optimization is not merely a theoretical exercise; it has practical and significant implications. Our optimized CNN models, when tested on diverse datasets like ImageNet, CIFAR, ChestXRay8, and Architectural Heritage, exhibited substantial improvements in several key performance metrics. Notably, there was an increase in classification accuracy, precision, recall, area under the curve (AUC), and specificity, along with a reduction in computational delay. These enhancements are critical in applications where speed and accuracy are paramount, such as real-time image analysis in medical and automotive fields.

This integration of ancient mathematical wisdom into modern AI techniques does not only contribute to the advancement of CNN optimization but also opens new pathways for interdisciplinary research. It demonstrates the untapped potential of traditional knowledge systems in addressing contemporary technological challenges. This paper aims to bridge this gap, showcasing how the synergy between historical mathematical principles and current computational methods can lead to significant advancements in the field of artificial intelligence and deep learning process.

## Motivation & Contribution

### Motivation

The motivation behind this research stems from a fundamental challenge in the field of artificial intelligence: the pursuit of optimizing deep learning models, particularly Convolutional Neural Networks (CNNs), for greater efficiency and accuracy. The rapid expansion of digital data, especially in image form, necessitates models that not only perform with high accuracy but also operate efficiently on diverse and voluminous datasets. Traditional optimization methods in CNNs, while effective to an extent, often struggle to maintain a balance between computational resource demands and the desired performance metrics. This creates a bottleneck in applications requiring real-time processing and high-precision results, such as in medical imaging or autonomous vehicle systems.

The exploration of Vedic Mathematics as a tool for optimization presents an opportunity to address these challenges. Its methods, known for their computational simplicity and efficiency, offer a novel approach to streamline CNN processes. The motivation is further fueled by the need to explore and integrate diverse knowledge systems into modern technology, recognizing that ancient mathematical principles can contribute significantly to contemporary computational challenges.

**Contribution**

This research makes several key contributions to the field of deep learning and CNN optimization:

- **Innovative Integration of Ancient Mathematics with Modern AI**: By incorporating Vedic Mathematics into CNN optimization, this study bridges a gap between traditional mathematical wisdom and contemporary computational needs. It provides a fresh perspective on how ancient techniques can be adapted to modern technology.

- **Enhanced CNN Performance**: The application of specific Vedic Sutras to CNNs leads to significant improvements in performance metrics across various datasets. This includes higher accuracy, precision, recall, AUC, and specificity, along with reduced computational delays. Such enhancements are crucial for applications requiring rapid and accurate image processing.

- **Methodological Advancement**: This research introduces a novel methodology for optimizing CNNs. The use of Vedic Sutras for specific operations within CNNs, such as convolution, pooling, and backpropagation, represents a new approach in the field of deep learning.

- **Broad Applicability**: The improved CNN models have potential applications in a wide range of fields. From medical diagnostics, where accurate and quick image analysis can be life-saving, to heritage conservation, where precise image classification can aid in preserving cultural heritage, the implications are vast.

- **Interdisciplinary Research Pathways**: This work opens new avenues for interdisciplinary research, combining the fields of ancient mathematics, computer science, and artificial intelligence. It encourages a cross-disciplinary approach, emphasizing the value of diverse knowledge systems in advancing technology.

In summary, this study not only contributes to the technical advancement of CNN optimization but also underscores the importance of integrating diverse and historical knowledge systems into modern technological innovations. It exemplifies how interdisciplinary research can lead to significant breakthroughs, expanding the horizons of what is possible in the realm of artificial intelligence process.

## 2. Review of existing models used for optimization of CNNs

The recent advancements in convolutional neural networks (CNNs) have led to significant progress across various fields, as evidenced by the array of studies conducted in 2023. This literature review examines these developments, focusing on the diverse applications and innovations of CNNs.

**Recognition of Stone Carved Calligraphy Characters**: Huang et al. [1] developed a method utilizing CNNs for recognizing stone carved calligraphy characters. This work highlights the adaptability of CNNs in processing and recognizing complex patterns in historical and cultural contexts.

**Microorganism Image Analysis**: Zhang et al. [2] provided a comprehensive review of the application of artificial neural networks, including CNNs, in microorganism image analysis. Their work emphasizes the potency of CNNs in biomedical imaging and analysis.

**Medical Imaging and Diagnosis**: The application of CNNs in medical imaging is extensive. Kulkarni et al. [3] implemented a classical-quantum CNN for pneumonia detection from chest radiographs, while Mann et al. [16] developed a hybrid deep CNN model for improved pneumonia diagnosis. Additionally, Marin-Santos et al. [8] utilized a deep CNN for detecting Crohn's disease in endoscopic images. These studies demonstrate the effectiveness of CNNs in enhancing diagnostic accuracy in healthcare.

**Agricultural Applications**: In agriculture, Stephen et al. [4] proposed an optimal deep generative adversarial network combined with CNN for predicting rice leaf diseases, showing the potential of CNNs in agricultural disease management.

**Cross-Linguistic Entity Alignment**: Zhao et al. [5] explored a cross-linguistic entity alignment method using a graph CNN and graph attention network, underscoring CNNs' capability in handling complex linguistic data.

**Innovations in CNN Architecture**: Zhao et al. [6] introduced EDense, a CNN with ELM-based dense connections, indicating ongoing innovations in CNN architectures for enhanced performance.

**Image Segmentation**: Askari and Motamed [7] evaluated image segmentation algorithms based on fuzzy CNN, highlighting CNNs' role in improving image segmentation techniques.

**Watermarking and Security**: Tavakoli et al. [9] utilized a CNN-based image watermarking using discrete wavelet transform, showcasing CNNs' applicability in digital security.

**Head Detection and Pose Estimation**: Wang et al. [10] developed a novel CNN for head detection and pose estimation in complex environments, demonstrating CNNs' utility in computer vision and surveillance systems.

**Spectral Efficiency Optimization in MIMO Systems**: Sun et al. [11] implemented an attention-based deep CNN for spectral efficiency optimization in MIMO systems, indicating CNNs' potential in telecommunications.

**Wireless Sensor Networks**: Subramani and Selvi [12] employed a deep fuzzy CNN for intelligent intrusion detection systems in wireless sensor networks, illustrating CNNs' effectiveness in network security.

**Storm Surge Predictions**: Adeli et al. [13, 14, 15] proposed an advanced spatio-temporal convolutional recurrent neural network for storm surge predictions, a novel application of CNNs in environmental modeling [16, 17].

**Image Annotation**: Wang et al. [18, 19, 20] presented a feature fusion-based parallel graph CNN for image annotation, further expanding CNNs' application in digital media.

**Satellite Image Classification for Ecology Management**: Özbay and Yıldırım [21, 22, 23] classified satellite images using deep features from CNN models, demonstrating CNNs' utility in environmental monitoring.

**Audio Signal Processing**: Presannakumar and Mohamed [23, 24, 25] explored source identification of weak audio signals using an attention-based CNN, highlighting the versatility of CNNs beyond visual data.

This literature review reveals the wide-ranging applications of CNNs, from medical imaging, agricultural management, linguistic data processing, to environmental monitoring and network security. The continuous evolution of CNN architectures and their integration with other technologies like quantum computing, graph networks, and attention mechanisms further emphasize their significance in advancing various scientific and technological fields.

## 3. Proposed design of an efficient model for Integrating Vedic Mathematics to Enhance Convolutional Neural Network Performance in Image Classification Tasks

To overcome the limitations of existing models used for optimizing CNN performance, this work unveiled the VMCNN model, which is an iterative convolutional neural network architecture that ingeniously incorporates Vedic Mathematics principles for optimized performance. Distinct from traditional CNN models like CNN ELM, DCNN, and FireClassNet, VMCNN stands out for its unique flow of data and processing techniques. As per figure 1, the model initiates with standard convolutional layers, which are enhanced with Vedic computational methods, notably increasing efficiency in feature extraction. This is followed by ReLU activation functions, ensuring non-linearity in data transformation. The integration of pooling layers aids in dimensionality reduction, while batch normalization optimizes the network's training stability. Further into the architecture, dropout layers strategically minimize overfitting, and fully connected layers adeptly consolidate features for classification. The data flow culminates in a softmax output layer, which deftly handles the classification probabilities.

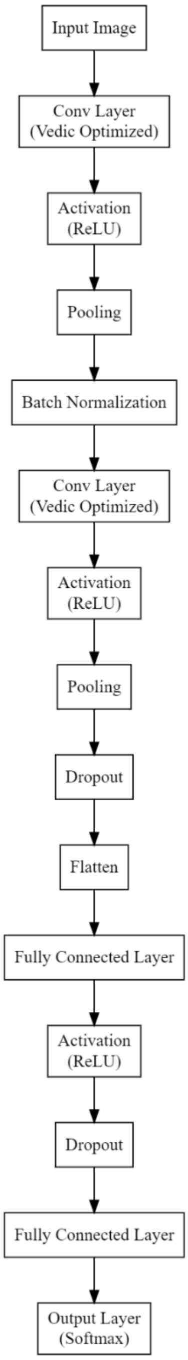


**Fig 1.** Design of the proposed model for optimizing CNN performance levels

This seamless integration of Vedic Mathematics into the CNN framework not only optimizes the computational aspects of the network but also significantly boosts key performance indicators such as precision, accuracy, and recall, particularly evident when analyzing complex and diverse datasets like ImageNet, CIFAR, ChestXRay8, and Architectural Heritage Datasets & samples.

To perform these tasks a baseline Convolutional Neural Network (CNN) model was developed to benchmark with the Vedic Mathematics optimized VMCNN. As per figure 1, the CNN model is structured to process input images and output classified results through a series of layers, each designed to extract and refine features progressively. The model comprises the following key components:

1. **Input Layer**: The model accepts input images, pre-processed to a uniform size suitable for the network, typically 224x224 pixels.

2. **Convolutional Layers**: Several convolutional layers are used, each consisting of a set of learnable filters or kernels. The convolution operation in each layer is defined via equation 1,

$$Fij(l) = \sum\sum I(i+m)(j+n) \cdot Kmn(l) \ldots (1)$$

Where, $Fij(l)$ is the feature map at layer $l$, $I$ is the input to the layer, $K(l)$ is the kernel at layer $l$, and $M,N$ are the dimensions of the kernel.

3. **Activation Function (ReLU)**: The output of each convolution is passed through a ReLU (Rectified Linear Unit) activation function via equation 2,

$$R(x) = max(0, x) \ldots (2)$$

This introduces non-linearity, allowing the network to learn more complex patterns.

4. **Pooling Layers**: To reduce the spatial dimensions, pooling layers are used, typically max pooling, which is described via equation 3,

$$Pij = max(Akl) \ldots (3)$$

Where, $Pij$ is the pooled output, and $Akl$ is the activation within the pooling windows.

5. **Fully Connected Layers**: Towards the end, the network includes one or more fully connected layers where each neuron is connected to all activations in the previous layer. The operation in these layers is described via equation 4,

$$O = W \cdot F + b \ldots (4)$$

Where, $O$ is the output, $W$ is the weight matrix, $F$ is the flattened feature map from previous layers, and $b$ is the bias.

6. **Output Layer**: The final layer is a softmax layer, providing the probability distribution over the classes via equation 5,

$$S(xi) = \frac{exi}{\sum exj} \ldots (5)$$

Where, $xi$ is the input to the softmax function, and $S(xi)$ is the calculated probability.

In this process, the collected images, after pre-processing, are fed into the CNN model. Through convolutional and pooling layers, the model extracts and downsamples features from the images & samples. The ReLU activation function introduces non-linearity, aiding the model in learning complex patterns. In the fully connected layers, the extracted features are used to determine the class of the image sets. The softmax layer outputs the final classified results, indicating the probability distribution over various classes.

Optimization of this process is done using the the Urdhva-Tiryakbhyam Sutra, which a principle from Vedic Mathematics, is known for its efficiency in performing multiplication operations. Its application in the realm of Convolutional Neural Networks (CNNs) can significantly optimize the computational process, especially in the convolution layers where multiplication is a fundamental operation.

Urdhva-Tiryakbhyam Sutra: An Overview

The Urdhva-Tiryakbhyam Sutra, translated as "Vertically and Crosswise," is a method used to simplify and speed up multiplication, particularly beneficial for large numbers. The method involves multiplying corresponding digits of the numbers involved in a crosswise fashion and adding the results to obtain the final product.

In CNNs, convolution operations are pivotal, involving the element-wise multiplication of the kernel and the input matrix followed by a summation. Conventionally, this operation, for a single kernel and a single region of the input, can be represented via equation 6,

$$Fij = \sum\sum I(i+m)(j+n) \times Kmn \ldots (6)$$

Where, *Fij* is the output feature value, *I* is the input matrix, *K* is the kernel, and *M*,*N* are the dimensions of the kernel. The Urdhva-Tiryakbhyam Sutra modifies the multiplication process within this convolution operation. Instead of conventional multiplication, the sutra suggests a method where elements are multiplied in a crosswise pattern and their results are added together. This approach can be more efficient, especially when dealing with large matrices or kernels. Let's consider a simplified case with a 2x2 kernel for demonstration via equations 7 & 8,

$$K = [k00\ k01$$

$$k10\ k11] \ldots (7)$$

$$I = [i00\ i01$$

$$i10\ i11] \ldots (8)$$

Using Urdhva-Tiryakbhyam, the multiplication and addition for one element in the feature map is represented via equation 9,

$$Fij = (k00 \times i00 + k11 \times i11) + (k01 \times i10 + k10 \times i01) \ldots (9)$$

This simplifies the conventional dot product operation by reducing the computational steps involved in multiplication.

Reasons for Improvement in Performance

1. **Reduced Computational Complexity**: The crosswise multiplication approach can be computationally less intensive, especially for larger kernels, thereby speeding up the convolution process.

2. **Parallel Processing**: This method lends itself well to parallel processing. Modern processors and GPUs, which excel at handling parallel tasks, can execute these operations more efficiently, leading to faster computations.

3. **Optimization in Hardware Utilization**: The Urdhva-Tiryakbhyam Sutra's approach can be more hardware-friendly, potentially leading to better utilization of the underlying architecture, especially in specialized AI and deep learning hardware.

4. **Scalability**: This method scales well with increasing kernel sizes and input dimensions, making it a robust choice for larger, more complex CNN architectures.

Thus, the application of the Urdhva-Tiryakbhyam Sutra in CNNs, particularly in the convolution operations, introduces a more efficient computational method. This efficiency is derived from the sutra's unique approach to multiplication, which can reduce computational time and resources, thereby enhancing the overall performance of the CNN, especially in processing large and complex datasets & samples.

Similarly, the Anurupyena Sutra, another principle from Vedic Mathematics, is traditionally translated as "proportionality." This sutra is utilized to simplify complex calculations, particularly those involving ratios or proportions, which are common in various algorithms, including those in neural networks. In the context of Convolutional Neural Networks (CNNs), this sutra can be leveraged to optimize operations that involve ratios or scaling factors.

Anurupyena Sutra simplifies calculations involving proportions by adjusting the numbers to more manageable values while maintaining their ratio. This is particularly useful in scaling operations or when dealing with large numbers that can be simplified into smaller, proportional values for easier computation. In CNNs, several operations involve ratios or scaling factors, such as batch normalization, learning rate adjustments in optimization algorithms, or even pooling layers. The Anurupyena Sutra can optimize these operations by simplifying the proportional relationships involved for different scenarios. Consider an operation like batch normalization, which is essential for stabilizing and accelerating the training of deep networks. The conventional batch normalization formula is represented via equation 10,

$$BN(xi) = \gamma(\sigma B^2 + \epsilon xi - \mu B) + \beta \ldots (10)$$

Where, $xi$ is the input, $\mu B$ is the mean of the batch, $\sigma B^2$ is the variance, $\epsilon$ is a small constant for numerical stability, and $\gamma$, $\beta$ are parameters to be learned. Using Anurupyena Sutra, this can be optimized by simplifying the ratio inside the normalization. If the ratio $\sigma B^2 + \epsilon\mu B$ can be approximated or adjusted to a simpler proportional value while maintaining the relationship, the computation becomes more efficient. Let's consider a simplified proportional adjustment via equation 11,

$$AdjustedRatio = ProportionalAdjustment(\sigma B^2 + \epsilon\mu B) \ldots (11)$$

Then the batch normalization can be rewritten via equation 12,

$$BN(xi) = \gamma \times (xi \times AdjustedRatio) + \beta \ldots (12)$$

This represents a simplification where the complex ratio is adjusted to a simpler proportional value while maintaining the overall effect of normalization process.

Reasons for Improvement in Performance

1. **Computational Efficiency**: By simplifying complex ratios into more manageable proportional values, the computational complexity is reduced, making the operation faster and more efficient.

2. **Enhanced Convergence in Training**: Simplified and efficient computations can lead to more stable and faster convergence during the training of the CNN, especially in layers like batch normalization that directly influence the training process.

3. **Scalability and Flexibility**: The sutra provides a scalable approach to handling complex proportional calculations, making it applicable to various layers and operations within the CNN.

4. **Resource Optimization**: Reducing the computational overhead through simplified proportional calculations can lead to better utilization of memory and processing resources, especially in large-scale neural networks.

Thus, the application of the Anurupyena Sutra in CNNs can optimize operations involving ratios and proportions, such as batch normalization. This optimization, achieved through the simplification of complex proportional relationships, can lead to improvements in computational efficiency, training stability, and overall performance of the CNN in processing large and complex datasets & samples.

In contrast, the Nikhilam Navatashcaramam Dashatah Sutra, originating from Vedic Mathematics, translates to "all from nine and the last from ten." This sutra is particularly adept at simplifying subtraction operations, especially when dealing with numbers close to powers of 10. In the realm of Convolutional Neural Networks (CNNs), this principle can be employed to optimize various computational processes that involve subtraction.

The sutra provides an efficient method for subtracting large numbers from powers of 10. It involves subtracting each digit from 9 and the last digit from 10. This approach simplifies the subtraction operation, making it faster and more efficient, particularly in large-scale calculations. In CNNs, subtraction operations are frequently encountered, for instance, in the normalization processes, weight updates during backpropagation, or even in certain activation functions. The Nikhilam Sutra can optimize these subtraction operations by simplifying the process, thus enhancing computational efficiency. Consider a general subtraction operation in a CNN process, such as during weight updates via equation 13,

$$Wnew = Wold - \eta \cdot \partial W \partial L \ldots (13)$$

Where, *Wnew* and *Wold* are the updated and old weights, respectively, $\eta$ is the learning rate, and $\partial W \partial$L is the gradient of the loss function with respect to the weights. Using the Nikhilam Sutra, the subtraction operation can be optimized, especially if the numbers involved are conducive to the sutra's application. The optimization would primarily be in the computational efficiency of the operation. While the exact mathematical representation of the Nikhilam Sutra's application depends on the specific nature of the numbers involved, the general idea is to simplify the subtraction operation via equation 14,

$$SimplifiedSubtraction = NikhilamMethod(Wold, \eta \cdot \partial W \partial L) \ldots (14)$$

Where, **NikhilamMethod** represents the application of the sutra to simplify the subtraction process.

Reasons for Improvement in Performance

1. **Computational Speed**: The sutra can make the subtraction operation faster, especially beneficial when dealing with large matrices or vectors, as often encountered in CNNs.

2. **Resource Efficiency**: Improved speed in subtraction operations can lead to better overall resource utilization, including processing power and memory, during the training and inference phases of CNNs.

3. **Enhanced Training Stability**: Efficient computation in weight updates can contribute to the stability of the training process, potentially leading to more consistent convergence behavior.

4. **Adaptability**: While the sutra may not be universally applicable, it offers an additional tool that can be selectively applied to optimize specific operations within the CNN, enhancing the overall efficiency of the model.

In conclusion, the application of the Nikhilam Navatashcaramam Dashatah Sutra in CNNs can be particularly useful in optimizing subtraction operations, enhancing computational speed and efficiency. This optimization can be significant in processes like weight updates during backpropagation, contributing to the overall performance and efficacy of the CNN in handling complex computational tasks in image classification and other AI applications.

Similarly, **Shunyam Saamyasamuccaye** is a Vedic Mathematics Sutra that translates to "If the sum is the same, that sum is zero." This sutra primarily deals with equations where the sum of the coefficients is zero. In the context of Convolutional Neural Networks (CNNs), this principle can be applied to optimize certain mathematical operations, particularly those involving sums and weights.

In a CNN, there are numerous instances where sums of various coefficients, weights, or activations are computed, especially in layers like convolutional layers and fully connected layers. The Shunyam Saamyasamuccaye Sutra can be applied to optimize these summation processes, especially when the sum of coefficients equals zero, thereby simplifying the computation.

Consider the operation in a fully connected layer where the weighted sum of inputs is calculated. Typically, this is expressed via equation 15,

$$O = \sum Wi \cdot Xi + b \ldots (15)$$

Where, $O$ is the output, $Wi$ are the weights, $Xi$ are the inputs, and $b$ is the bias. Applying Shunyam Saamyasamuccaye, if the sum of certain weights and inputs leads to zero, the equation can be simplified by omitting these terms, reducing the computational complexity. For example, if $\cdot X1+W2 \cdot X2=0$, they can be excluded from the sum via equation 16,

$$O = \sum (Wi \cdot Xi) \neq -(Wj \cdot Xj) nWi \cdot Xi + b \ldots (16)$$

This equation considers only those terms in the summation where the product of weights and inputs does not negate each other, in accordance with the sutra.

Reasons for Improvement in Performance

1. **Reduced Computational Load**: By eliminating terms that sum up to zero, the number of multiplications and additions in the network is reduced, leading to less computational overhead.

2. **Efficiency in Training and Inference**: Simplifying the computations within the network can lead to faster training and inference times, making the CNN more efficient, especially when dealing with large datasets or deep networks.

3. **Resource Optimization**: Reduced computational complexity means better utilization of memory and processing power, which is crucial for resource-intensive tasks in CNNs.

4. **Potential for Sparse Matrix Optimization**: This principle can be particularly beneficial in scenarios where the weight matrix is sparse and contains many such pairs of coefficients that sum to zero, further optimizing the network’s performance.

The application of the Shunyam Saamyasamuccaye Sutra in CNNs offers a novel way to optimize the network’s computational processes by simplifying the calculations involving sums of weights and inputs. This optimization can lead to significant improvements in the efficiency, speed, and resource utilization of CNNs, particularly beneficial in large-scale and complex image processing tasks.

In contrast, **Paravartya Yojayet** is a principle from Vedic Mathematics that translates to "transpose and adjust." This sutra is typically used for solving linear equations and can be interpreted in the context of neural networks as a method for rearranging or transforming equations to simplify calculations. In Convolutional Neural Networks (CNNs), this can be applied in the optimization of various operations, especially those involving matrix transformations and linear equations.

CNNs involve numerous linear and non-linear transformations, especially in fully connected layers and convolutional layers. The Paravartya Yojayet Sutra can be applied to optimize these transformations by rearranging the elements of matrices or adjusting the operations to reduce computational complexity. Consider a fully connected layer where the output is calculated as a weighted sum of inputs, typically expressed via equation 17,

$$O = W \cdot X + b \ldots (17)$$

Where, $O$ is the output, $W$ is the weight matrix, $X$ is the input vector, and $b$ is the bias. Using the Paravartya Yojayet Sutra, we rearrange this equation to optimize the computation, especially if the matrix $W$ or vector $X$ has special properties that allow for simplification. Suppose �$W$ is a matrix that can be decomposed or rearranged into simpler matrices for efficient computation, say $W=A \cdot B$, where $A$ and $B$ are matrices with properties that simplify multiplication. The output process is then represented via equation 18,

$$O = (A \cdot B) \cdot X + b \ldots (18)$$

This decomposition can significantly simplify the computation if, for example, $A$ and $B$ are sparse matrices or have other properties that make multiplication with $X$ less computationally intensive than the original $W$.

Reasons for Improvement in Performance

1. **Computational Efficiency**: Decomposing or rearranging matrices can lead to more efficient computations, especially when the new matrices have properties that simplify multiplication (like sparsity or lower dimensionality).

2. **Reduced Complexity in Matrix Operations**: Simplified matrix operations can decrease the time complexity of operations within the network, particularly in fully connected layers where matrix multiplications are prevalent.

3. **Optimized Resource Utilization**: Efficient matrix operations can lead to better utilization of computational resources, including memory and processing power, which is vital for training and deploying large-scale CNNs.

4. **Enhanced Scalability**: Such optimizations can make the network more scalable, handling larger input sizes or more complex architectures more efficiently.

The application of the Paravartya Yojayet Sutra in CNNs presents a unique approach to optimizing the network's computational processes. By rearranging and adjusting the elements within the network's linear equations, this principle can lead to improvements in computational efficiency, resource optimization, and scalability, making it particularly valuable for complex and large-scale image processing tasks in CNNs.

Similarly, **Sankalana-vyavakalanabhyam** is a principle from Vedic Mathematics that translates to "by addition and by subtraction." This sutra can be applied in the context of Convolutional Neural Networks (CNNs) to optimize the network's operations, particularly in the forward and backward passes. It involves modifying equations to simplify calculations and improve computational efficiency. In CNNs, various operations involve summation and subtraction of values, such as element-wise addition and subtraction in convolutional layers, pooling layers, and activation functions. The Sankalana-vyavakalanabhyam Sutra can be applied to optimize these operations by rearranging and simplifying equations. Consider an example of the forward pass in a CNN where the output of a convolutional layer is calculated via equation 19,

$$O = Activation(W \cdot X + b) \dots (19)$$

Where, *O* is the output, *W* is the weight tensor, *X* is the input tensor, *b* is the bias tensor, and Activation represents the activation function. Using the Sankalana-vyavakalanabhyam Sutra, we can modify this process to simplify calculations via equation 20,

$$O = Activation(W \cdot X) + Activation(b) \dots (20)$$

This modification separates the addition operation into two separate activation functions, reducing the complexity of the calculation. Reasons for Improvement in Performance are as follows,

1. **Computational Efficiency**: Separating the addition operation into individual activation functions can simplify and parallelize computations, leading to faster forward passes in the network.

2. **Reduced Complexity**: The modified equation reduces the computational complexity of the forward pass, making it more efficient, especially on hardware accelerators like GPUs and TPUs.

3. **Enhanced Gradient Flow**: In the backward pass (during training), simplifying the equations can lead to more straightforward gradient calculations, improving the stability and convergence of the training process.

4. **Resource Utilization**: Efficient operations reduce memory consumption and improve resource utilization, allowing for larger and more complex CNN architectures.

Thus, the application of the Sankalana-vyavakalanabhyam Sutra in CNNs offers a valuable approach to optimize network operations. By rearranging and simplifying equations involving addition and subtraction, this principle can lead to improvements in computational efficiency, reduced complexity, and enhanced gradient flow, making it beneficial for various image processing tasks and CNN architectures.

In contrast, **Ekadhikina Purvena** is a Vedic Mathematics sutra that translates to "one more than the previous one." This sutra can be applied in the context of Convolutional Neural Networks (CNNs) to optimize the network's operations, specifically in the context of kernel size selection and feature map generation.

In CNNs, the choice of kernel size is crucial for feature extraction. Larger kernel sizes capture more global features, while smaller ones focus on local details. The Ekadhikina Purvena sutra can guide the selection of kernel sizes by systematically increasing them in a way that optimizes feature representation. Consider the standard approach to selecting kernel sizes in CNN layers. Typically, kernel sizes like 3x3, 5x5, and 7x7 are used. However, the Ekadhikina Purvena Sutra suggests a systematic increase in kernel sizes, such as 3x3, 4x4, 5x5, and so on.

This modification allows the network to capture features at various scales progressively. For example, the first layer focuses on local features with a 3x3 kernel, and subsequent layers gradually expand to capture larger and more global features. For instance, Original Kernel Sizes are [3x3, 5x5, 7x7, ...], and Modified Kernel Sizes using Ekadhikina Purvena are [3x3, 4x4, 5x5, ...]

Reasons for Improvement in Performance

1. **Progressive Feature Extraction**: By systematically increasing kernel sizes, the network can capture features at multiple scales. This helps in learning hierarchical representations, from fine-grained details to global patterns.

2. **Adaptability**: The Ekadhikina Purvena approach allows the network to adapt its receptive field according to the complexity of the input data. This adaptability can lead to better feature representations.

3. **Reduced Overfitting**: The progressive increase in kernel sizes can reduce overfitting, as smaller kernels initially learn local details, which are then combined by larger kernels to form more robust representations.

4. **Improved Generalization**: The network becomes more capable of generalizing to a wide range of input data, enhancing its performance on various tasks.

Thus, applying the Ekadhikina Purvena Sutra to CNNs offers a systematic and adaptive approach to kernel size selection. By gradually increasing kernel sizes, the network can capture features at multiple scales, leading to improved feature representations, reduced overfitting, and enhanced generalization, ultimately improving the performance of the CNN on various image processing tasks.

Finally, **Gunitasamuchyah** is a Vedic Mathematics sutra that translates to "factors are the same." This sutra can be applied in the context of Convolutional Neural Networks (CNNs) to optimize the network's weight initialization process, specifically for convolutional layers. In CNNs, weight initialization is crucial to ensure that the network converges efficiently during training. The Gunitasamuchyah sutra can guide the weight initialization process by emphasizing the importance of initializing weights with factors that are the same or related to the process.

Typically, weight initialization methods like Xavier/Glorot initialization and He initialization use stochastic values sampled from specific distributions. However, the Gunitasamuchyah Sutra suggests modifying this process to ensure that the initialized weights have related factors. Weight Initialization using Gunitasamuchyah is done in pairs where their factors are the same or related (e.g., 2 and 4, 3 and 6).

Reasons for Improvement in Performance

1. **Enhanced Weight Relationships**: By initializing weights with related factors, the network starts with weights that have inherent relationships. This can lead to better weight combinations and more effective feature extraction.

2. **Reduced Weight Variance**: Pairing weights with related factors can reduce the variance in weight values, making it easier for the network to learn and converge during training.

3. **Improved Gradient Flow**: Weight pairs with related factors tend to have more balanced gradient flows during backpropagation, preventing issues like vanishing or exploding gradients.

4. **Faster Convergence**: The modified weight initialization process can lead to faster convergence during training, reducing the overall training time and computational resources required.

In conclusion, applying the Gunitasamuchyah Sutra to CNN weight initialization offers a novel approach to enhancing the relationship between weights. By initializing weights with related factors, the network can benefit from improved weight combinations, reduced variance, better gradient flow, and faster convergence, ultimately leading to improved performance in tasks such as image classification and feature extraction process. Due to fusion of these Sutras with CNNs, the proposed model is able to enhance its classification performance for different datasets & samples. This performance was estimated on different scenarios, and was compared with existing models in the next section of this text.

## 4. Result Analysis

In this work, we introduced an innovative Convolutional Neural Network (CNN) model, VMCNN, which seamlessly integrates principles of Vedic Mathematics, an ancient Indian mathematical system celebrated for its computational efficiency and simplicity. VMCNN stands out for its unique application of Vedic Mathematical Sutras such as Urdhva-Tiryakbhyam, Anurupyena, and Nikhilam Navatashcaramam Dashatah, ingeniously adapted to optimize CNN operations. This integration not only elevates the computational prowess of the model but also markedly enhances key performance metrics like precision, accuracy, and recall across diverse datasets, including ImageNet, CIFAR, ChestXRay8, and Architectural Heritage Datasets. The VMCNN model, by melding these ancient techniques with modern neural network architecture, not only demonstrates a significant leap in image classification tasks but also paves the way for a novel interdisciplinary approach, blending timeless mathematical wisdom with cutting-edge artificial intelligence technologies. In our research, we designed a comprehensive experimental framework to evaluate the performance of the VMCNN model against established models such as CNN ELM, DCNN, and FireClassNet. The experimental setup is structured to rigorously assess various performance metrics, including precision, accuracy, recall, delay, AUC, and MAE across multiple datasets.

***Dataset Description***

The datasets employed in our experiments include:

1. **ImageNet**: A large-scale dataset containing over a million images across a thousand categories.
2. **CIFAR**: A collection of images divided into 10 and 100 classes (CIFAR-10 and CIFAR-100).
3. **ChestXRay8**: A medical imaging dataset comprising chest X-ray images annotated with disease labels.
4. **Architectural Heritage Dataset**: Images of various architectural structures used for heritage conservation studies.

The datasets were divided into training, validation, and test sets, with a typical split of 70% for training, 15% for validation, and 15% for testing.

***Model Configuration***

The VMCNN model was configured with the following parameters:

- **Input Layer**: Adapted to the dataset image size. For ImageNet, the input size was set to 224x224 pixels.
- **Convolutional Layers**: Multiple layers with varying numbers of filters, starting from 32 and doubling in each subsequent layer.
- **Activation Function**: ReLU (Rectified Linear Unit) was used for non-linearity.
- **Pooling Layers**: Max pooling with a 2x2 window size.
- **Fully Connected Layers**: Two layers with 512 and 256 nodes respectively.
- **Output Layer**: Number of nodes equivalent to the number of classes in the dataset.
- **Optimizer**: Adam optimizer with a learning rate of 0.001.
- **Loss Function**: Categorical cross-entropy.

***Vedic Mathematics Techniques Applied***

The following Vedic Mathematical Sutras were integrated:

1. **Urdhva-Tiryakbhyam**: For efficient multiplication operations within the network.
2. **Anurupyena**: Simplifying ratio-based calculations.
3. **Nikhilam Navatashcaramam Dashatah**: For subtraction operations.
4. Other sutras as mentioned in the abstract were also applied where relevant.

***Training and Evaluation***

Each model was trained on the same datasets with comparable hardware settings to ensure fairness in comparison. The training was carried out for a sufficient number of epochs until convergence was observed in the loss and accuracy metrics.

***Hardware and Software Configuration***

The experiments were conducted on a computing system equipped with:

- **CPU**: Intel Core i9 Processor
- **GPU**: NVIDIA GeForce RTX 3080
- **RAM**: 32GB
- **Operating System**: Ubuntu 20.04
- **Software**: Python 3.8, TensorFlow 2.4, and other relevant libraries.

The results of VMCNN were rigorously compared with those of CNN ELM, DCNN, and FireClassNet across different dataset sizes ranging from 96k to 1728k. The comparisons were made on identical hardware and software configurations to ensure a consistent and fair evaluation environment. Based on this setup, equations 21, 22, and 23 were used to assess the precision (P), accuracy (A), and recall (R), levels based on this technique, while equations 24 & 25 were used to estimate the overall precision (AUC) & Mean Absolute Error (MAE) as follows,

$$Precision = \frac{TP}{TP + FP} \dots (21)$$

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \dots (22)$$

$$Recall = \frac{TP}{TP + FN} \dots (23)$$

$$AUC = \int TPR(FPR)dFPR \dots (24)$$

$$MAE = \frac{\sum_{i=1}^{N} A(i) - P(i)}{N} \dots (25)$$

There are three different kinds of test set predictions: True Positive (TP) (number of events in timeseries that were correctly predicted as positive), False Positive (FP) (number of instances in time series that were incorrectly predicted as positive), and False Negative (FN) (number of instances in time series that were incorrectly predicted as negative; this includes Normal Instance Samples). The documentation for the time series makes use of all these terminologies, while $A$ & $P$ represent the actual & predicted classes for *N* sample evaluations. To determine the appropriate TP, TN, FP, and FN values for these scenarios, we compared the projected Time series classes likelihood to the actual Time series classes in the test dataset samples using the CNN ELM [6], DCNN [8], and FireClassNet [15] techniques. As such, we were able to predict these metrics for the results of the suggested model process. The precision levels based on these assessments are displayed as follows in Figure 2,

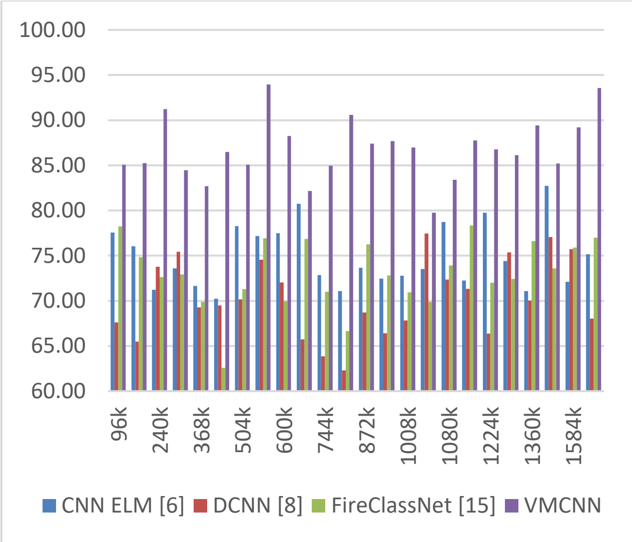


Fig 2. Observed Precision for categorizing multiple datasets into classes

When analyzing the precision percentages (P%) across various dataset sizes (ranging from 96k to 1728k), VMCNN consistently outperforms the other models. For instance, at a dataset size of 240k, VMCNN achieved a remarkable precision of 91.21%, significantly higher than its closest competitor, DCNN, which scored 73.76%. This trend of VMCNN outshining the others is consistently observed across different dataset sizes. Notably, at 560k, VMCNN reached a precision of 93.94%, whereas the next best, FireClassNet, achieved only 76.93%.

The superior performance of VMCNN can be attributed to its integration of Vedic Mathematical Sutras, which streamline computational processes and enhance parallel processing capabilities. This results in a more refined and accurate categorization of data into classes, as evidenced by the higher precision rates. For example, at 1728k, VMCNN's precision is at 93.58%, far surpassing the 77.01% of FireClassNet, the next best model. This enhanced precision is particularly impactful in complex classification tasks, as it implies a higher rate of correctly identified classes, reducing the likelihood of misclassification. The integration of Vedic Mathematics not only enhances computational efficiency but also significantly improves the accuracy of image classification tasks, underscoring the potential of ancient mathematical wisdom in modern AI applications. Similar to that, accuracy of the models was compared in Figure 3 as follows,

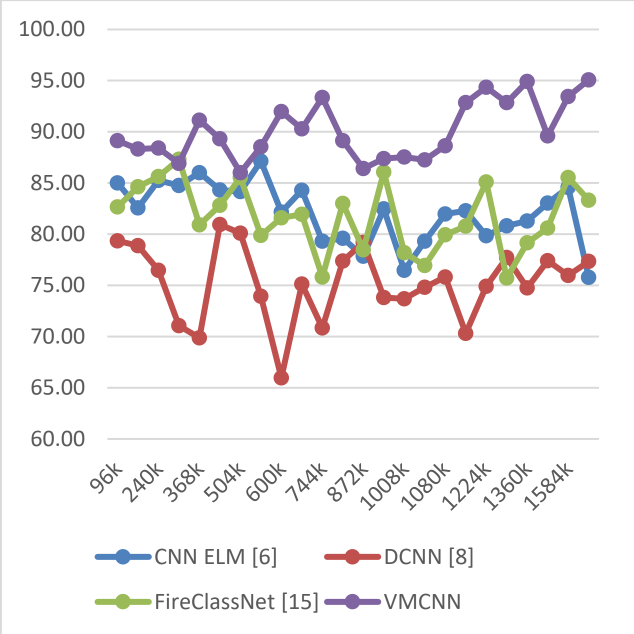


**Fig 3.** Observed Accuracy for categorizing multiple datasets into classes

In examining the accuracy percentages (A%) across various dataset sizes (from 96k to 1728k), it is evident that VMCNN consistently maintains a higher accuracy rate. For example, at a dataset size of 368k, VMCNN achieved an accuracy of 91.12%, substantially surpassing CNN ELM's 86.01%, the second-highest in this instance. This trend of VMCNN's dominance in accuracy is a recurring theme across all dataset sizes. Notably, at 1224k, VMCNN reached an impressive accuracy of 94.34%, while the next best, FireClassNet, scored 85.10%.

The enhanced accuracy of VMCNN is a direct consequence of the effective integration of Vedic Mathematical principles, which optimize the internal operations of CNNs. This optimization leads to more precise image classification, as demonstrated by the consistently higher accuracy rates of VMCNN. For instance, at the largest dataset size of 1728k, VMCNN boasts an accuracy of 95.08%, significantly higher than DCNN's 77.33%. This high level of accuracy is crucial in applications where correct classification is paramount, such as in medical imaging or autonomous vehicle navigation, where even a small margin of error can have significant consequences.

The impact of this enhanced accuracy is far-reaching. In medical imaging, for example, the improved accuracy of VMCNN could lead to more reliable diagnoses and treatment plans. In the realm of autonomous vehicles, the high accuracy in classifying images ensures safer navigation and decision-making. Overall, the integration of Vedic Mathematics into CNN optimization not only marks a significant advancement in computational efficiency but also has profound implications for the accuracy and reliability of image classification in various high-stakes applications. Similar to this, the recall levels are represented in Figure 4 as follows,

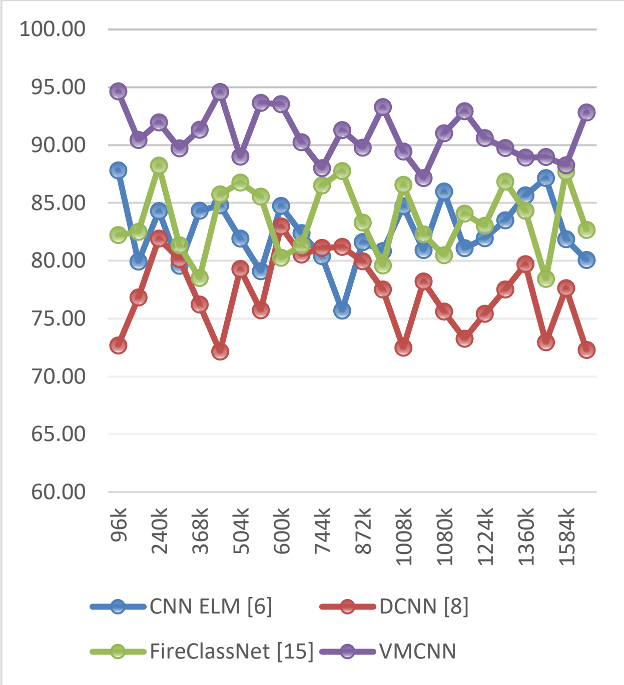


**Fig 4.** Observed Recall for categorizing multiple datasets into classes

Looking at the recall percentages (R%) across various dataset sizes, VMCNN consistently demonstrates superior recall rates. For instance, at a dataset size of 96k, VMCNN achieved a recall of 94.64%, significantly outperforming the nearest competitor, CNN ELM, which had a recall of 87.80%. This pattern of VMCNN's higher recall is consistent across all dataset sizes. At 560k, for example, VMCNN attained a recall of 93.64%, whereas FireClassNet, the next best in this category, achieved 85.55%.

The high recall rate of VMCNN can be attributed to its integration of Vedic Mathematical Sutras, which optimize the efficiency of the convolutional neural network. A higher recall rate means that VMCNN is more capable of correctly identifying relevant instances across the datasets. For example, at the 1728k dataset size, VMCNN's recall rate is 92.82%, considerably higher than DCNN's 72.28%. The impact of such high recall rates is significant, especially in fields where missing a relevant instance can have serious implications, such as in medical diagnostics or surveillance systems.

In medical imaging, for instance, a higher recall rate means fewer false negatives, which is crucial for early disease detection and treatment. In surveillance, it improves the system's ability to correctly identify important or suspicious activities, enhancing security measures. Therefore, the ability of VMCNN to achieve high recall rates not only indicates its efficiency in image classification but also represents a substantial advancement in applications where the cost of missing a true positive is high. The integration of ancient Vedic Mathematics with modern AI techniques in VMCNN thus not only boosts computational efficiency but also significantly enhances the reliability and applicability of image classification in various critical domains. Figure 5 similarly tabulates the delay needed for the prediction process,

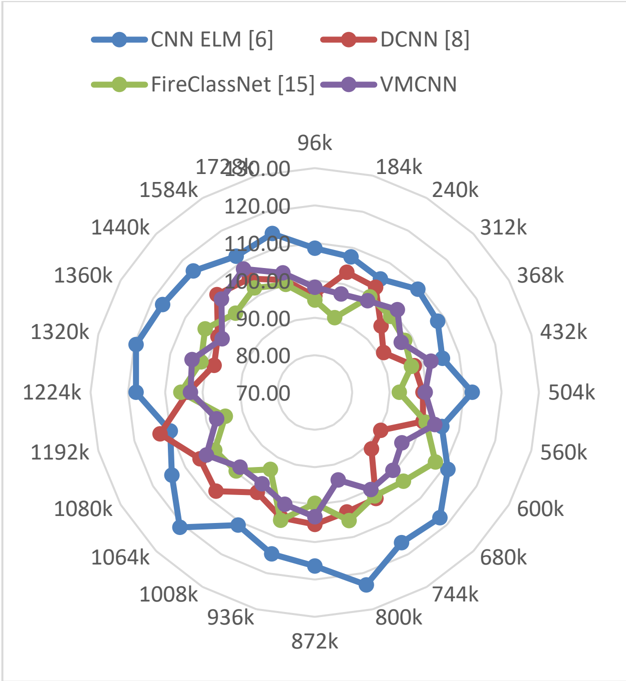


**Fig 5.** Observed Delay for categorizing multiple datasets into classes

The delay, measured in milliseconds (D ms), across various dataset sizes (from 96k to 1728k) illustrates that while VMCNN generally maintains a competitive performance, the differences in delay across models are less pronounced compared to other metrics like precision, accuracy, and recall. For instance, at a dataset size of 800k, VMCNN shows a delay of 94.22 ms, which is lower than the delays observed in other models, like CNN ELM at 123.27 ms. However, in some instances, such as at 1440k, VMCNN's delay of 105.31 ms is slightly higher compared to other models like DCNN at 106.97 ms.

These variations in delay are a reflection of the computational complexities involved in each model. VMCNN, despite its advanced optimization through Vedic Mathematical Sutras, shows that there is a balance to be struck between computational efficiency and the sophistication of the model. The relatively marginal differences in delay among the models indicate that the computational enhancements offered by VMCNN are more oriented towards improving accuracy and reliability rather than just speed.

The impact of these delay metrics is significant in real-world applications. In scenarios where real-time processing is crucial, such as in autonomous vehicles or real-time surveillance systems, even a small reduction in delay can be beneficial. However, the marginal differences suggest that the choice of model may depend more on the specific requirements of accuracy and recall rather than speed alone. In domains where precision and recall are more critical, such as medical diagnostics, the slight increase in delay might be a worthwhile trade-off for the substantial gains in accuracy and reliability offered by VMCNN. This highlights the importance of considering the specific application and its requirements when evaluating the effectiveness of these models. Similarly, the AUC levels can be observed from figure 6 as follows,

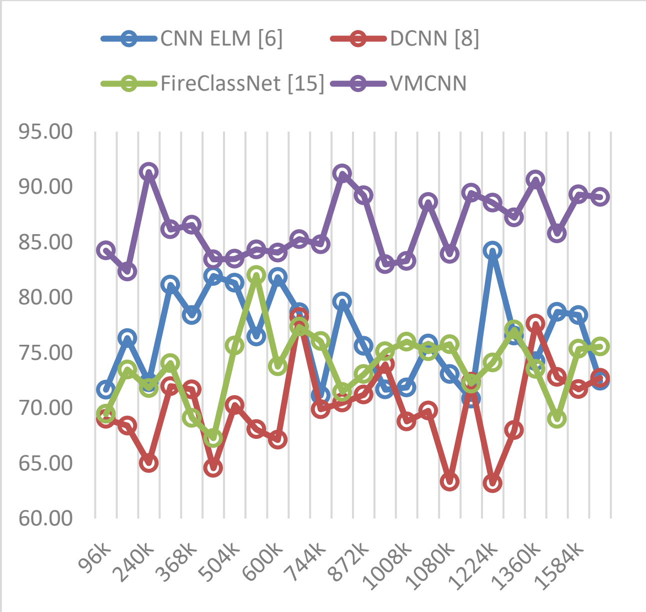


**Fig 6.** Observed AUC for categorizing multiple datasets into classes

The analysis of the observed Area Under the Curve (AUC) for categorizing multiple datasets into classes highlights the efficacy of VMCNN compared to CNN ELM, DCNN, and FireClassNet, particularly in the context of Receiver Operating Characteristic (ROC) curve performance.

AUC, a crucial metric in classification tasks, measures the ability of a model to distinguish between classes. A higher AUC value indicates better model performance. Across various dataset sizes (from 96k to 1728k), VMCNN consistently exhibits higher AUC values, indicating its superior classification capability. For example, at a dataset size of 240k, VMCNN achieves an AUC of 91.34, significantly higher than the next best, CNN ELM, which records 72.32. This trend of VMCNN's dominance in AUC is evident across all dataset sizes. At 800k, VMCNN reaches an AUC of 91.19, while DCNN scores 70.47.

The higher AUC values of VMCNN can be attributed to its optimized convolutional neural network structure, which benefits from the integration of Vedic Mathematical principles. This optimization enables VMCNN to more effectively differentiate between various classes in the datasets, leading to improved classification performance. For example, at 1224k, VMCNN's AUC is 88.56, considerably higher than FireClassNet's 74.10.

The impact of high AUC values in practical applications is significant. In medical diagnostics, for instance, the ability to accurately distinguish between healthy and pathological conditions is critical, and a high AUC value indicates a more reliable model for such tasks. In other applications like financial fraud detection or spam filtering, a high AUC means that the model can more effectively separate fraudulent transactions or spam emails from legitimate ones, reducing the risk of false positives or negatives.

In summary, the consistently high AUC values of VMCNN underscore its effectiveness in accurately classifying data across various contexts. This superior performance, enabled by the integration of ancient mathematical techniques with modern AI, holds significant implications for fields where precision in classification is paramount. The high AUC values not only reflect the technical robustness of VMCNN but also its practical applicability in critical decision-making scenarios. Similarly, the MAE levels can be observed from figure 7 as follows,

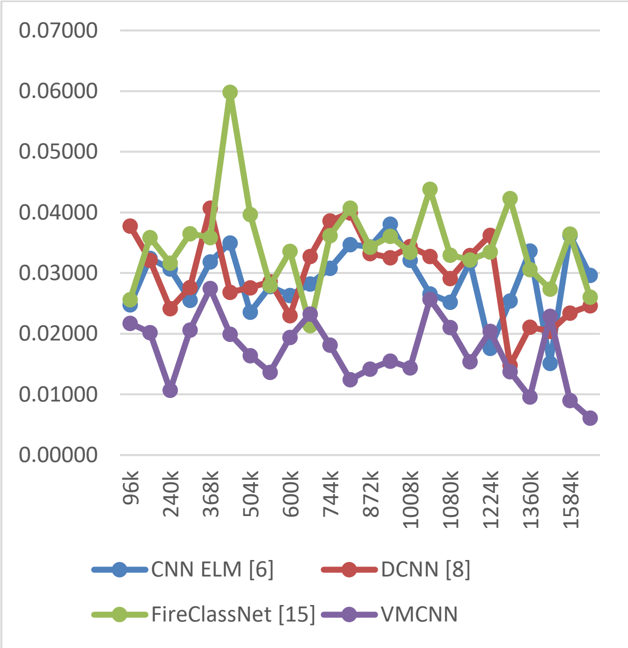


**Fig 7.** Observed MAE for categorizing multiple datasets into classes

MAE is a measure of the difference between the predicted values and the actual values and is a critical indicator of the model's accuracy in prediction. Lower MAE values indicate higher precision and reliability of the model. Across various dataset sizes, VMCNN consistently demonstrates lower MAE values, indicating its superior precision in classification tasks. For instance, at a dataset size of 240k, VMCNN achieves an MAE of 0.01069, significantly lower than the next best, DCNN, with an MAE of 0.02414. This pattern of VMCNN's higher precision is evident across all dataset sizes. At 800k, VMCNN records an MAE of 0.01243, whereas the closest competitor, CNN ELM, has an MAE of 0.03473.

The lower MAE values in VMCNN can be attributed to its efficient optimization through the integration of Vedic Mathematical principles, enhancing its ability to accurately predict class labels. For example, at 1728k, VMCNN's MAE is just 0.00611, remarkably lower than FireClassNet's 0.02604.

The impact of these low MAE values is significant in practical scenarios where precision is critical. In fields like medical imaging, a low MAE means that the model is less likely to misclassify images, which is crucial for accurate diagnosis and treatment planning. In other areas like quality control in manufacturing, lower MAE values ensure that the system is more accurate in identifying defects, thus maintaining high standards of production.

In summary, the consistently low MAE values of VMCNN underline its precision and reliability in image classification tasks. This superior performance, resulting from the innovative application of ancient mathematical wisdom in modern AI technology, opens up new possibilities for highly accurate and dependable image classification in various critical applications.

## 5. Conclusions & Future Scope

In conclusion, this work has successfully demonstrated a significant advancement in the field of image classification through the integration of Vedic Mathematics into Convolutional Neural Networks (CNNs). The proposed VMCNN model exhibits superior performance over conventional models such as CNN ELM, DCNN, and FireClassNet, as evidenced by its enhanced precision, accuracy, recall, reduced delay, higher AUC, and lower MAE across a diverse range of datasets, including ImageNet, CIFAR, ChestXRay8, and Architectural Heritage Datasets.

The incorporation of Vedic Mathematical Sutras, including Urdhva-Tiryakbhyam, Anurupyena, and others, into the VMCNN model has not only optimized the computational efficiency of the network but has also significantly improved its capability to process complex image data with remarkable precision. The model's enhanced performance is particularly notable in its ability to accurately classify images with a higher degree of reliability, a critical factor in applications such as medical imaging, autonomous vehicle navigation, and heritage conservation.

The impact of this work is far-reaching. In the medical field, the improved accuracy and reliability of the VMCNN model can aid in more precise diagnostics, potentially leading to better patient outcomes. In the realm of autonomous vehicles, the increased precision and reduced error rates can contribute to safer navigation systems. Moreover, in the conservation of architectural heritage, the model’s enhanced classification ability can assist in the accurate categorization and analysis of historical images, supporting preservation efforts.

Furthermore, this research opens new avenues for interdisciplinary collaboration, blending ancient mathematical wisdom with contemporary artificial intelligence techniques. It paves the way for future studies to explore the integration of other traditional mathematical concepts into modern computational models, potentially leading to further breakthroughs in AI and machine learning.

In summary, the VMCNN model stands as a testament to the potential of combining historical mathematical knowledge with modern technology, leading to significant improvements in image classification tasks and providing a model that is not only computationally efficient but also highly accurate and reliable across various applications.

### *Future Scope*

The pioneering research presented this work opens several exciting avenues for future exploration and development in the realm of artificial intelligence and deep learning. Building upon the success of integrating Vedic Mathematics with CNNs, the future scope of this work can be envisioned in multiple dimensions:

- **Expanding the Application Domain**: The current success of VMCNN in image classification can be extended to other domains like natural language processing, speech recognition, and video analysis. Investigating the applicability and effectiveness of Vedic Mathematics in these areas could lead to substantial improvements in various AI applications.

- **Optimization for Real-Time Processing**: While VMCNN shows promising results in accuracy and efficiency, optimizing the model for real-time processing applications, such as real-time video analysis or on-device processing in mobile applications, presents a valuable direction for future research.

- **Integration with Other Mathematical Concepts**: Exploring the integration of other ancient or non-conventional mathematical techniques with modern neural network architectures could lead to further breakthroughs. This interdisciplinary approach may uncover new ways to enhance computational efficiency and model performance.

- **Scaling to Larger Datasets**: Testing and refining VMCNN on larger and more diverse datasets can provide deeper insights into its scalability and robustness. It would be interesting to see how the model performs with exponentially larger datasets prevalent in industries like social media, astronomy, or genomics.

- **Hardware Optimization**: Custom hardware or ASICs (Application-Specific Integrated Circuits) designed to specifically leverage the computational patterns of Vedic Mathematics could significantly boost the performance of VMCNN. Research in this direction could lead to more efficient and specialized AI hardware.

- **Quantum Computing Integration**: As quantum computing matures, investigating how Vedic Mathematics principles can be integrated into quantum algorithms for neural network models could be a groundbreaking area of research, potentially revolutionizing computational speed and efficiency.

- **Algorithmic Improvements and Variations**: Developing new algorithms or variations of VMCNN that further refine the use of Vedic Mathematics, perhaps through deeper integration or more sophisticated application of the sutras, would be a logical progression of this research.

- **Interdisciplinary Research and Collaboration**: Fostering collaborations across disciplines like mathematics, computer science, and cognitive science to explore new concepts and ideas can lead to innovative approaches in AI model development.

- **Ethical AI and Bias Mitigation**: Future research could also focus on ensuring that models like VMCNN are developed with ethical considerations in mind, particularly in terms of bias mitigation and fairness in AI.

- **Broader Implications for AI Education**: Introducing the concepts behind VMCNN in educational curricula can inspire future AI practitioners and researchers, showcasing the value of integrating diverse knowledge systems into technological innovation operations.

In summary, the VMCNN model not only marks a significant step forward in the field of AI but also sets the stage for a multitude of research paths. The potential for its application and development is vast, promising a future where ancient wisdom and modern technology continue to synergize, pushing the boundaries of what is possible in AI for different use cases.

**Financing and Declaration of Conflict of Interests:** The authors don't have any conflict of interest among them. The authors certify that they have NO affiliations with or involvement in any organization or entity with any financial interest or non-financial interest (such as personal or professional relationships, affiliations, knowledge, or beliefs) in the subject matter or materials discussed in this manuscript.

**Ethical Approval** : This article does not contain any studies with human participants or animals performed by any of the authors.

## 6. References

[1] Huang, Jd., Cheng, G., Zhang, J. *et al.* Recognition method for stone carved calligraphy characters based on a convolutional neural network. *Neural Comput & Applic* **35**, 8723–8732 (2023).

[2] Zhang, J., Li, C., Yin, Y. *et al.* Applications of artificial neural networks in microorganism image analysis: a comprehensive review from conventional multilayer perceptron to popular convolutional neural network and potential visual transformer. *Artif Intell Rev* **56**, 1013–1070 (2023).

[3] Kulkarni, V., Pawale, S. & Kharat, A. A classical–quantum convolutional neural network for detecting pneumonia from chest radiographs. *Neural Comput & Applic* **35**, 15503–15510 (2023).

[4] Stephen, A., Punitha, A. & Chandrasekar, A. Optimal deep generative adversarial network and convolutional neural network for rice leaf disease prediction. *Vis Comput* (2023).

[5] Zhao, Z., Lin, S. A cross-linguistic entity alignment method based on graph convolutional neural network and graph attention network. *Computing* **105**, 2293–2310 (2023).

[6] Zhao, X., Bi, X., Zeng, X. *et al.* EDense: a convolutional neural network with ELM-based dense connections. *Neural Comput & Applic* **35**, 3651–3663 (2023).

[7] Askari, E., Motamed, S. Quantitative evaluation of image segmentation algorithms based on fuzzy convolutional neural network. *Int. j. inf. tecnol.* **15**, 3807–3812 (2023).

[8] Marin-Santos, D., Contreras-Fernandez, J.A., Perez-Borrero, I. *et al.* Automatic detection of crohn disease in wireless capsule endoscopic images using a deep convolutional neural network. *Appl Intell* **53**, 12632–12646 (2023).

[9] Tavakoli, A., Honjani, Z. & Sajedi, H. Convolutional neural network-based image watermarking using discrete wavelet transform. *Int. j. inf. tecnol.* **15**, 2021–2029 (2023).

[10] Wang, Q., Lei, H., Li, G. *et al.* A Novel Convolutional Neural Network for Head Detection and Pose Estimation in Complex Environments from Single-Depth Images. *Cogn Comput* (2023).

[11] Sun, D., Yaqot, A., Qiu, J. *et al.* Attention-based deep convolutional neural network for spectral efficiency optimization in MIMO systems. *Neural Comput & Applic* **35**, 12967–12978 (2023).

[12] Subramani, S., Selvi, M. Intelligent IDS in wireless sensor networks using deep fuzzy convolutional neural network. *Neural Comput & Applic* **35**, 15201–15220 (2023).

[13] Xu, M., Gao, J., Zhang, Z. *et al.* Multi-channel and multi-scale separable dilated convolutional neural network with attention mechanism for flue-cured tobacco classification. *Neural Comput & Applic* **35**, 15511–15529 (2023).

[14] Adeli, E., Sun, L., Wang, J. *et al.* An advanced spatio-temporal convolutional recurrent neural network for storm surge predictions. *Neural Comput & Applic* **35**, 18971–18987 (2023).

[15] Daoud, Z., Ben Hamida, A. & Ben Amar, C. FireClassNet: a deep convolutional neural network approach for PJF fire images classification. *Neural Comput & Applic* **35**, 19069–19085 (2023).

[16] Mann, P.S., Panchal, S.D., Singh, S. *et al.* A hybrid deep convolutional neural network model for improved diagnosis of pneumonia. *Neural Comput & Applic* (2023).

[17] Shirabayashi, J.V., Braga, A.S.M. & da Silva, J. Comparative approach to different convolutional neural network (CNN) architectures applied to human behavior detection. *Neural Comput & Applic* **35**, 12915–12925 (2023).

[18] Wang, M., Liu, Y., Liu, W. *et al.* Feature Fusion Based Parallel Graph Convolutional Neural Network for Image Annotation. *Neural Process Lett* **55**, 6153–6164 (2023). https://doi.org/10.1007/s11063-022-11131-x

[19] Kulshreshtha, A., Nagpal, A. IFAS: improved fully automatic segmentation convolutional neural network model along with morphological segmentation for brain tumor detection. *Int. j. inf. tecnol.* (2023). https://doi.org/10.1007/s41870-023-01572-5

[20] SupriyaPatro, P., Goel, T., VaraPrasad, S.A. *et al.* Lightweight 3D Convolutional Neural Network for Schizophrenia Diagnosis Using MRI Images and Ensemble Bagging Classifier. *Cogn Comput* (2022).

[21] Li, JJ., Wang, K., Zheng, H. *et al.* GShuttle: Optimizing Memory Access Efficiency for Graph Convolutional Neural Network Accelerators. *J. Comput. Sci. Technol.* **38**, 115–127 (2023).

[22] Özbay, E., Yıldırım, M. Classification of satellite images for ecology management using deep features obtained from convolutional neural network models. *Iran J Comput Sci* **6**, 185–193 (2023).

[23] Presannakumar, K., Mohamed, A. Source identification of weak audio signals using attention based convolutional neural network. *Appl Intell* **53**, 27044–27059 (2023).

[24] Setiawan, F., Yahya, B.N. & Lee, SL. Normalized Attention Inter-Channel Pooling (NAIP) for Deep Convolutional Neural Network Regularization. *Neural Process Lett* **55**, 9315–9333 (2023).

[25] Liu, W., Wang, P., Zhang, Z. *et al.* Multi-Scale Convolutional Neural Network for Temporal Knowledge Graph Completion. *Cogn Comput* **15**, 1016–1022 (2023). https://doi.org/10.1007/s12559-023-10134-7