

# Matrix Representations of CNN Layers: A Linear Algebra Approach to Efficient Training

Nanduri Srinivas<sup>1</sup>, Allacheruvu Brahmaiah<sup>2</sup>, Nunna Gayatri Devi<sup>3</sup>, Madasu Ramanjani Devi<sup>4</sup>,  
Kunapuli Venkata Rama Rao<sup>5</sup>

<sup>2</sup>Department of Computer Science, RK College of Engineering, Vijayawada, Andhra Pradesh, India.

<sup>1,5</sup>Department of Mathematics, RK College of Engineering, Vijayawada, Andhra Pradesh, India.

Department of Mathematics, S.R.K. Institute of Technology, Vijayawada, Andhra Pradesh, India.

**Abstract -** In this paper, the authors propose to apply a mathematical concept in Python that they believe we will encounter in our careers in data science and machine learning. Data, Machine Learning Module, and Training Module are the three primary pillars of the machine learning paradigm. Here are some links between machine learning and mathematics: Algorithms for machine learning are developed using notions from linear algebra. It links the machine learning algorithm of a Convolution Neural Network to be used on an enormous number of datasets to investigate how matrices, norms, and vectorisation techniques are required to represent and manipulate data in convolutional neural networks (CNNs), therefore increasing image processing efficiency. The ideal approach to becoming acquainted with mathematical principles in machine learning algorithms is better understood thanks to our goal. It makes mathematical comprehension essential and makes it possible to develop machine-learning solutions for issues in the in daily life situation actual world.

**Keywords:** Linear Algebra, Vectors, Matrices, Accuracy, feature map, Convolutional Neural Network.

## I. INTRODUCTION

The machine learning of a machine to think and behave like a person without having to be explicitly programmed to do so is known as machine learning. Without realising it, we already apply machine learning in our daily lives. Machine learning is being used for email spam detection, spell checking, and even the suggestion of the YouTube video that got you here [1]. Algorithms are used in machine learning to learn tasks; these algorithms are fed data and trained to execute these tasks. This implies that we don't need to rewrite our application as data changes over time; instead, the authors propose that it identify trends and absorb fresh information. The science of giving robots human-like intelligence so they can sense, reason, act, and adapt is known as artificial intelligence, of which machine learning is a subset. A subfield of machine learning called "deep learning" is motivated by the way the human brain functions. We're getting closer to a day when machines can think and learn thanks to machine learning [2]. In machine learning, model selection refers to the process of selecting the most appropriate model for a given issue. The process of choosing a model is influenced by several variables, including the task, dataset, and model type. It will

be solely working on developing a new machine model for mathematics in this effort. Linear algebra, which includes vectors, matrices, norms, and broadcasting, is the foundation of mathematics [4], [5].

Linear Algebra is one use of artificial intelligence (AI) that gives systems the capacity to automatically learn from experience and get better at it without explicit programming, which is machine learning of Convolution Neural Network. Computer programs that can access data and utilise it to learn for themselves are developed using machine learning. One area of mathematics that has applications in both science and engineering is linear algebra.[4]. It aids with the comprehension and operation of machine learning algorithms, followed by the steps of linear algebraic operations. A straightforward machine learning method developed using machine learning expertise. Vectors, Norms and Matrices[3] will be dealt with in linear algebra. Since all computations rely on this type of vector, Norms and matrices.

Machine learning algorithms for image analysis leverage various filters to enhance the efficiency and accuracy of the image processing tasks. Filters are applied at different stages to preprocess image data, such as noise reduction, edge detection, and contrast enhancement, which improve the clarity of features critical for model performance. Convolutional Neural Networks (CNNs) are a popular architecture for image recognition.[7], utilise convolutional layers to automatically learn and apply complex filters that capture patterns like textures, shapes, and edges. [6], [7], [8], [9], [10].

## II LITERATURE SURVEY

Linear algebra is a fundamental component of mathematics that focuses on structures that can be closed using addition and scalar multiplication. It comprises fundamental theories about linear equations, matrices, determinants, vector spaces, and linear transformations. This topic is critical to a wide range of applications, including mathematical physics, engineering, and healthcare, with a focus on machine learning.[13]. Linear algebra, which uses vectors, matrices, and linear transformations, is important in machine learning because it allows computers to make independent judgments.[4]. The ideas of linear algebra, notably those relating vector spaces and transformations, are critical for dealing with systems of linear equations, which are vital to solving many practical issues, particularly in machine learning.[14] Furthermore, deep learning, a subset of machine learning that uses artificial neural networks with multiple processing layers, employs both linear algebra and calculus to facilitate the automated extraction of complex features from data, as seen in methods for identifying fruits from images.[1]. This connection exemplifies the critical relationship between linear algebra and machine learning, underlining the need to understand linear algebraic ideas for advancement in computational techniques.

## II. MACHINE LEARNING PROCESS

In order to create a precise model, we have organised our work into multiple stages. Fig. 1. is represents the data evaluation process. These actions are

### 1. Gathering data

In light of our issue description, we attempt to gather information from various sources at this stage. Acquiring both high-quality and quantity of data is crucial.

### 2. Prepare data

We ensure that our data is streamlined and refined. Such that the data can be processed and understood by the ML model with ease. We must divide the data into testing and training sets.

### 3. Choose a model

The right and best model for our project must be chosen depending on the kind of data we have.

### 4. Model training

All of the data that we have is divided into the training set so that our machine learning model can comprehend and derive insights from it.

### 5. Prediction

Estimating a number that is absent or unknown using historical or present data.

### 6. Evaluation

With the assistance of the test set of data, we assess our trained machine learning model. We utilise the remaining test set of data to assess our ML model and determine its accuracy.

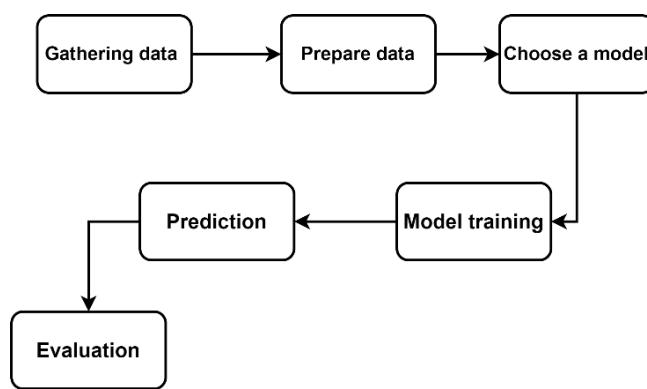


Fig. 1. Data evolution process

## III LINEAR ALGEBRA MODEL

Linear algebra of Figure 2 represents the applications of linear algebra, which uses vectors, matrices, norms, and vectorisation operations to describe data and relationships, and is especially useful in signal processing, image recognition, and natural language processing applications, to evaluate trends in data, reduce dimensionality, and enhance computing performance.

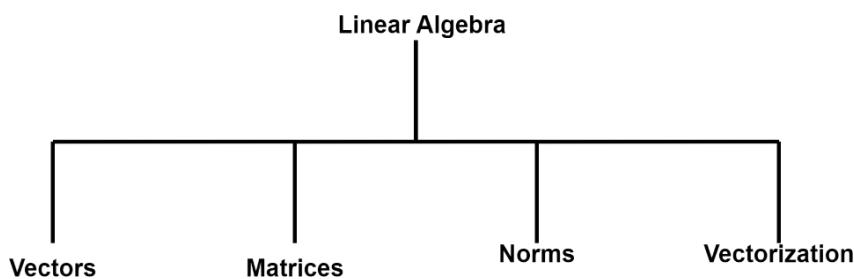


Fig. 2. Application of Linear Algebra

### A. Vectors

One-dimensional arrays of integers are called vectors, and they are commonly represented using lowercase, bold, and italics. Because vectors are ordered, an element may be retrieved using its index. Vectors are used to represent specific points in space. Two-length vectors indicate a place in a 2D matrix, three-length vectors indicate a location in a 3D cube, and n-length vectors indicate a location in an n-dimensional tensor.

Here are some basic Vector examples[15].

(i). One-dimensional vector

$$[V]$$

(ii). Two-Dimensional vector

$$\begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix}$$

## B. Matrices

A matrix is a carefully ordered group of items. Any number or function can be one of its components, and each distinct piece of data, a number, symbol, or expression, is referred to as a matrix element. The number of columns and rows in a matrix varies.

Here are some basic Matrix examples

$$\text{i) } \begin{bmatrix} a & h & q & p \\ h & b & f & q \\ g & f & c & r \end{bmatrix}$$

$$\text{ii) } \begin{bmatrix} 1 & i & -2i \\ i & -2 & 14 \\ -2i & 14 & 3 \end{bmatrix}$$

$$\text{iii) } \begin{bmatrix} 21 & 14 \\ 13 & 3 \end{bmatrix}$$

## C. Norms

Norms are a measure of the vector's size; they quantify the simple (Euclidean) distance from the origin. The most widely used norm in machine learning is  $\|x\|$ , which may be used in place of  $\|x\|_2$ .

Here are some basic Norms examples[11].

If  $\alpha = (a_1, a_2, a_3) \in V_3(R)$  then the three-dimensional Euclidean space of Norm or the length of the vector  $\alpha$  written:  $\|\alpha\| = \sqrt{a_1^2 + a_2^2 + a_3^2}$

## D. Vectorisation

A method for accelerating computations is called vectorisation, and it is particularly useful for deep learning algorithms where large amounts of data must be processed in a variety of ways. One method would be to transform the data into vectors and matrices rather than using an explicit loop and doing scalar calculations. The

Fig. 3. Representation of a basic Vectorisation example.

If  $\beta = \{1,2,3,4,5,6\}$  are the elements of  $V_3(R)$  then the tensor is

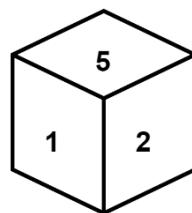


Fig. 3. Basic Vectorisation example

#### IV. NEURAL NETWORKS FOR LINEAR ALGEBRA

Linear algebra is fundamental to many scientific areas and is used extensively in machine learning and data science applications. Recent breakthroughs in deep learning have resulted in unique techniques for analysing massive datasets and identifying complicated patterns. CNNs, well-known for their image-processing skills, have intrinsic qualities that make them appropriate for linear algebra issues. Figure 4 expresses the working of a convolutional neural network. This study tries to bridge the gap between classic linear algebra methods and machine learning models, proving the usefulness of CNNs in executing fundamental linear operations.[16].

#### 1. Convolution Neural Network (CNN)

- Suitable For Image data, Spatial data, and tasks where the local structure of the data is important.
- Mathematical operations: CNNs use convolution operations, which involve sliding a filter (kernel) across the input data and applying matrix multiplication. This results in feature maps that capture local Patterns.

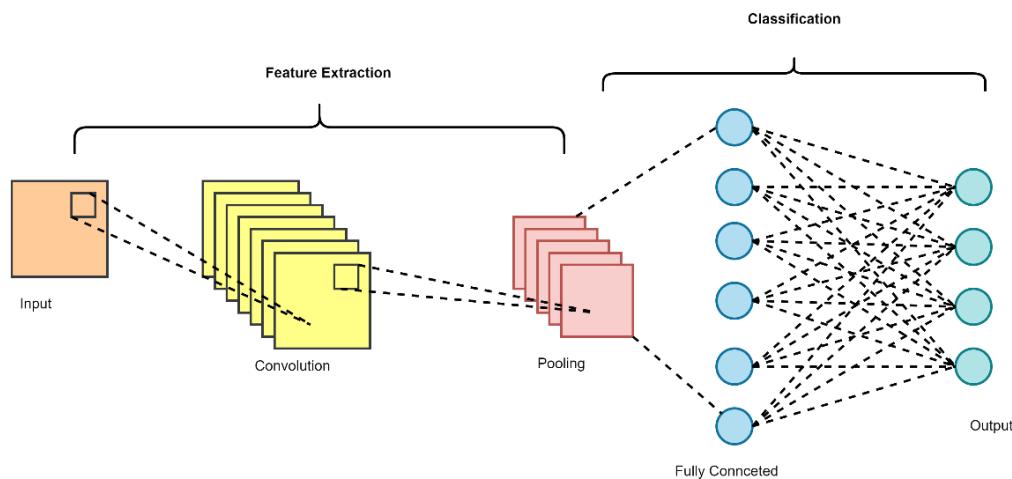


Fig. 4. Architecture of Convolution Neural Network

In Image classification, a convolution operation on an input image tensor (height  $\times$  width  $\times$  channel) with a filter (kernel) tensor produces a feature map. The operation is

$$\text{Feature Map} = \text{Input Tensor} \times \text{kernel}$$

Here  $\times$  denotes the convolution operation, which involves multiplying the kernel matrix with overlapping sections of the input tensor and summing the results[2]. In Convolutional Neural Networks (CNNs), the values 1, 0, and -1 often appear in kernel matrices (filters) used for edge detection. These values help

highlight specific features by emphasising differences in pixel intensity, such as vertical, horizontal, or diagonal edges, enhancing feature extraction for image analysis.

In below these are a few examples of basic CNN models for image classification using linear algebra.

### 1.1 Matrix Operations in Convolutional Neural Network

Imagine we have a grayscale image of size  $4 \times 4$  Pixels and we want to apply a  $2 \times 2$  Filter to detect a feature in this image.

Steps:

#### 1. Input Image

\*The image is represented as  $4 \times 4$  matrices

$$\text{Input} = \begin{bmatrix} 1 & 2 & 3 & 0 \\ 4 & 5 & 6 & 1 \\ 7 & 8 & 9 & 0 \\ 1 & 2 & 3 & 4 \end{bmatrix}$$

#### 2. Filter (kernel)

- $A_{2 \times 2}$  filter is used to detect a specific feature

$$\text{Filter} = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$$

#### 3. Convolution Operation

- The filter slides over the image, performing element-wise multiplication and summing the results at each position.
- 1<sup>st</sup> position (top-left corner)

$$\text{Input slice} = \begin{bmatrix} 1 & 2 \\ 4 & 5 \end{bmatrix}$$

$$\text{Filter} * \text{Input slice} = (1 \times 1) + (2 \times 0) + (4 \times 0) + (5 \times -1)$$

$$= 1 - 5 = -4$$

- 2<sup>nd</sup> position (Moving right)

$$\text{Input slice} = \begin{bmatrix} 2 & 3 \\ 5 & 6 \end{bmatrix}$$

$$\text{Filter} * \text{Input slice} = (2 \times 1) + (3 \times 0) + (5 \times 0) + (6 \times -1)$$

$$= 2 - 6 = -4$$

- 3<sup>rd</sup> position (Moving right)

$$\text{Input slice} = \begin{bmatrix} 3 & 0 \\ 6 & 1 \end{bmatrix}$$

$$\text{Filter*Input slice} = (3 \times 1) + (0 \times 0) + (6 \times 0) + (1 \times -1)$$

$$= 3 - 1 = 2$$

➤ 4<sup>th</sup> position (Moving right)

$$\text{Input slice} = \begin{bmatrix} 4 & 5 \\ 7 & 8 \end{bmatrix}$$

$$\text{Filter*Input slice} = (4 \times 1) + (5 \times 0) + (7 \times 0) + (8 \times -1)$$

$$= 4 - 8 = -4$$

➤ 5<sup>th</sup> position (Moving right)

$$\text{Input slice} = \begin{bmatrix} 5 & 6 \\ 8 & 9 \end{bmatrix}$$

$$\text{Filter*Input slice} = (5 \times 1) + (6 \times 0) + (8 \times 0) + (9 \times -1)$$

$$= 5 - 9 = -4$$

➤ 6<sup>th</sup> position (Moving right)

$$\text{Input slice} = \begin{bmatrix} 6 & 1 \\ 9 & 0 \end{bmatrix}$$

$$\text{Filter*Input slice} = (6 \times 1) + (1 \times 0) + (9 \times 0) + (0 \times -1)$$

$$= 6 - 1 = 5$$

➤ 7<sup>th</sup> position (Moving right)

$$\text{Input slice} = \begin{bmatrix} 7 & 8 \\ 1 & 2 \end{bmatrix}$$

$$\text{Filter*Input slice} = (7 \times 1) + (8 \times 0) + (1 \times 0) + (2 \times -1)$$

$$= 7 - 2 = 5$$

➤ 8<sup>th</sup> position (Moving right)

$$\text{Input slice} = \begin{bmatrix} 8 & 9 \\ 2 & 3 \end{bmatrix}$$

$$\text{Filter*Input slice} = (8 \times 1) + (9 \times 0) + (2 \times 0) + (3 \times -1)$$

$$= 8 - 3 = 5$$

➤ 9<sup>th</sup> position (Moving right)

$$\text{Input slice} = \begin{bmatrix} 9 & 0 \\ 3 & 4 \end{bmatrix}$$

$$\text{Filter*Input slice} = (9 \times 1) + (0 \times 0) + (3 \times 0) + (4 \times -1)$$

$$= 9 - 4 = 5$$

#### 4. Resulting feature map

After sliding the filter over the entire image, we get  $A_{3 \times 3}$  feature map

$$\text{Feature Map} = \begin{bmatrix} -4 & -4 & 2 \\ -4 & -4 & 5 \\ 5 & 5 & 5 \end{bmatrix}$$

### 1.2 Tensor Operations in Convolutional Neural Network

Consider an RGB image of size 4\*4 pixels, where each pixel has three colour channels (Red, Green, Blue).

The image is represented as a 3D tensor steps:

#### 1. Input tensor

The input image is represented as a 3D tensor with dimensions (*Height*  $\times$  *Width*  $\times$  *Channels*):

Input Tensor = (*Height*  $\times$  *Width*  $\times$  *Channels*):

Assume the input tensor looks like this

$$\text{Input tensor} = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 0 & 1 & 2 & 1 & 2 & 1 & 2 \\ 1 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 2 & 1 & 2 & 1 & 2 & 1 & 0 & 1 & 0 & 1 \end{bmatrix}$$

#### 2. Filter (Kernel)

A  $3 \times 3$  The filter is applied to the input tensor. This filter has the same depth as the input tensor channels, and its sides are over the height and width of the tensor.

Assume the filter is

$$\text{Filter Tensor} = [[1 \ 0 \ -1 \ 0 \ 1 \ 0 \ -1 \ 0 \ 1 \\ 0 \ 1 \ 0 \ 1 \ 0 \ -1 \ 0 \ -1 \ 0 \\ -1 \ 0 \ 1 \ 0 \ -1 \ 0 \ 1 \ 0 \ -1]]$$

This filter will detect specific patterns across all these channels simultaneously.

#### 3. Convolutional operation

The filter slides over the input tensor, performing element-wise multiplication across all three channels and summing the results to produce a single number at each position.

First position (top-left corner):

$$\text{Input slice} = [[2 \ 1 \ 0 \ 1 \ 0 \ 2 \ 0 \ 1 \ 0 \\ 1 \ 0 \ 1 \ 2 \ 1 \ 0 \ 1 \ 0 \ 2]]$$

Result = Sum of element-wise multiplications between the filter and input slice

The convolution operation continues as the filter slides across the entire tensor, calculating the result at each position.

#### 4. Resulting feature map (Output tensor)

After applying the filter across the entire tensor and the input tensor, the result is the 2D feature map

Assume the resulting feature map looks like this:

$$\text{Feature map} = \begin{bmatrix} 5 & 2 \\ -3 & 4 \end{bmatrix}$$

This feature map represents the deleted features in the original image.

### 1.3 Vector Operations in Convolutional Neural Networks

#### 1. Input vector

- Assume we have a 1D input vector of length 8:

$$\text{Input } [1,3,2,4,6,5,7,9]$$

#### 2. Filter (Kernel)

- A 1D filter (Kernel) of length 3 is issued to extract features from the input vector. The filter conducted specific patterns, like trends or a 0 in the data.

$$\text{Filter} = [1,0,-1]$$

- This filter is designed to detect changes in the data.

#### 3. Convolution operations

- The filter slides over the input vector, performing element-wise multiplication and summing the results at each position.

1. 1<sup>st</sup> Position

$$\text{Input slice} = [1,3,2]$$

$$\text{Convolution Result} = (1 \times 1) + (3 \times 0) + (2 \times -1) = 1 + 0 - 2 = -1$$

2<sup>nd</sup> Position

Input slice=[3,2,4]

Convolution Result = $(3X1) + (2X0) + (4X - 1) = 3 + 0 - 4 = -1$

3<sup>rd</sup> Position

Input slice=[2,4,6]

Convolution Result = $(2X1) + (4X0) + (6X - 1) = 2 + 0 - 6 = -4$

4<sup>th</sup> Position

Input slice=[4,6,5]

Convolution Result = $(4X1) + (6X0) + (5X - 1) = 4 + 0 - 5 = -1$

5<sup>th</sup> position

Input slice=[6,5,7]

Convolution Result = $(6X1) + (5X0) + (7X - 1) = 6 + 0 - 7 = -1$

6<sup>th</sup> position

Input slice=[5,7,9]

Convolution Result = $(5X1) + (7X0) + (9X - 1) = 5 + 0 - 9 = -4$

#### 4. Resulting Feature Map (Output vector)

After applying the filter across the input vector, the resulting feature map (Output vector) is,

Feature Map = [-1, -1, -4, -1, -1, -4]

This output vector captures the local patterns detected by the filter.

#### V. RESULTS AND OBSERVATIONS

Here's a structured approach for presenting results and observations:

1. Combining linear algebra, matrix operations, and vectorisation enhances machine learning, especially for complex calculations. Matrices and norms in CNNs improve efficiency and accuracy, aiding large dataset representation in image processing.
2. Using vectorization, CNNs efficiently handle large datasets, reducing computational demands while preserving accuracy. Matrix operations replace loops, accelerating data processing and cutting down model training time.
3. Efficient data handling in CNNs, enabled by advanced matrix operations, enhances pattern recognition, making CNNs effective for applications like facial recognition, object detection, and medical imaging in large-scale datasets.
4. A strong foundation in mathematics is crucial for data scientists and machine learning experts, enabling robust algorithm development, enhanced model performance, and practical solutions in fields like healthcare, finance, and autonomous systems.

## VI. CONCLUSION

Mathematical principles, particularly linear algebra, play an important role in the development and comprehension of machine learning algorithms. Investigating the connections between mathematics and machine learning emphasises the use of matrices, norms, and vectorisation techniques, particularly in Convolutional Neural Networks (CNNs). These principles are critical for efficiently encoding and processing huge datasets, hence improving the performance and applicability of image-processing tasks in machine learning. A good understanding of fundamental mathematical concepts serves as a foundation for developing powerful machine-learning models that can solve real-world issues. Understanding the mathematical foundations of machine learning not only helps to build efficient algorithms but also enhances problem-solving skills in the data science and machine learning fields.

## VII. REFERENCES

- [1] Sahar Halim, "Application of Linear Algebra in Machine Learning," Feb. 2020, [Online]. Available: <https://www.ijert.net/archives/V7/i2/IRJET-V7I2687.pdf>
- [2] H. A. Markus Götz, "Machine Learning-Aided Numerical Linear Algebra: Convolutional Neural Networks for the Efficient Preconditioner Generation," 2018, doi: DOI: 10.1109/ScalA.2018.00010.
- [3] J. N. S. A. R. Vasishtha, *Linear Algebra*. Krishna Prakashan Media. [Online]. Available: [https://books.google.co.in/books?id=jM6M18axJ3QC&source=gb\\_s\\_navlinks\\_s](https://books.google.co.in/books?id=jM6M18axJ3QC&source=gb_s_navlinks_s)

- [4] Dr K. Abinisha. K, "MACHINE LEARNING APPROACH FOR LINEAR ALGEBRA," no. 05/May-2022, [Online]. Available: [https://www.irjmets.com/uploadedfiles/paper//issue\\_5\\_may\\_2022/24011/final/fin\\_i\\_rjmets1653474126.pdf](https://www.irjmets.com/uploadedfiles/paper//issue_5_may_2022/24011/final/fin_i_rjmets1653474126.pdf)
- [5] S. L. Campbell and C. W. Gear, "The index of general nonlinear DAES," *Numer Math*, vol. 72, no. 2, pp. 173–196, 1995.
- [6] C. Hamburger, "Quasimonotonicity, regularity and duality for nonlinear systems of partial differential equations," *Ann Mat Pura Appl*, vol. 169, no. 2, pp. 321–354, 1995.
- [7] Avijeet Biswal, "Convolutional Neural Network Tutorial," Nov. 2023. [Online]. Available: <https://www.simplilearn.com/tutorials/deep-learning-tutorial/convolutional-neural-network#:~:text=A%20convolutional%20neural%20network%20is%20used%20to%20detect%20and%20classify,of%20any%20convolutional%20neural%20network.>
- [8] S. N. PK Mittal, *Matrices*. Chand Publishing, 2010. [Online]. Available: [https://books.google.co.in/books?id=CfzLwAEACAAJ&source=gbs\\_navlinks\\_s](https://books.google.co.in/books?id=CfzLwAEACAAJ&source=gbs_navlinks_s)
- [9] K. H. RAY KUNZE, *LINEAR ALGEBRA*. [Online]. Available: <https://www.cin.ufpe.br/~jrs1/Books/Linear%20Algebra%20-20Kenneth%20Hoffman%20%26%20Ray%20Kunze%20.pdf>
- [10] Stephen H Friedberg, *Linear Algebra*. [Online]. Available: <https://anandinstitute.org/pdf/lenearal.pdf>
- [11] B. P.B.jain, *Basic Linear Algebra*. 1995. [Online]. Available: <https://mdu.ac.in/UpFiles/UpPdfFiles/2020/Jan/BASIC%20ABSTRACT%20ALG EBRA.pdf>
- [12] Sai Mannam, *A Primer on Linear Algebra in Machine Learning*. 2021. [Online]. Available: <https://www.jyi.org/2021-october/2021/10/27/a-primer-on-linear-algebra-in-machine-learning>
- [13] S. M. Dr. Shankar Nayak, "Using Linear Algebra to solve the Machine Learning Application", [Online]. Available: <https://www.ijaresm.com/using-linear-algebra-to-solve-the-machine-learning-application>
- [14] A. F. Ghulam Farid, "APPLICATION OF CALCULUS AND LINEAR ALGEBRA IN DEEP LEARNING FOR FRUIT IDENTIFICATION FROM ITS IMAGE".
- [15] S. L. Campbell and C. W. Gear, "The index of general nonlinear DAES," *Numer Math*, vol. 72, no. 2, pp. 173–196, 1995.
- [16] L. A. Jinglan Zhang, "Review of deep learning: concepts, CNN architectures, challenges, applications, future directions," *SpringerOpen*, Mar. 2021.