IMAGE CLASSIFICATION OF CIFAR-10 DATASET

Tejas Bhavsar

Dr D. Y. Patil School of Science &

Technology, Dr D. Y. Patil Vidyapeeth,

Pune, India

bhavsartejas027@gmail.com

Rohan Mashere

Dr D. Y. Patil School of Science &

Technology, Dr D. Y. Patil Vidyapeeth,

Pune, India

rohanmashere2@gmail.com

Prathmesh Nangare

Dr D. Y. Patil School of Science &

Technology, Dr D. Y. Patil Vidyapeeth,

Pune, India

prathmeshnangare19@gmail.com

Akanksha goel

Dr D. Y. Patil School of Science &

Technology, Dr D. Y. Patil Vidyapeeth,

Pune, India

akansha.rkgit@gmail.com

Abstract-- In recent years, image classification has been challenging in part to the lack of technique available for image classification. Image classification is referred to as the classification of images among many preset categories. Several studies taken place to tackle the problems in image classification, however the results were confined to a basic low-level image. The deep learning neural network such as Convolutional neural networks (CNNs) and machine learning algorithms such as support vector machines (SVMs) are used for image classification, with kernels such as linear, radial basis function (RBF), and polynomial kernels and a penalty parameter, C, of the error term ranging from 0.0001 to 100. In this study, we will implement image classification using CNN. CNN is a deep neural network that is mostly used for visual imaging the process. CNN is implemented using a multilayer perceptron that works on a hierarchical model of network development before delivering to a fully linked layer. This layer connects all of the neurons and processes their output. This study shows how to categorize CIFAR-10 datasets with a convolutional neural network (CNN). To extract the underlying information from images, learnable filters and pooling layers were used. To avoid overfitting and improve accuracies during validation and testing, dropout, regularization, and convolution variation were utilized. This deeper network improved test accuracy and reduced overfitting. CIFAR-10 is an extremely popular computer vision database. The collection of dataset consists of 60,000 images grouped into 10 target classes, with every group comprising 6,000 images of 32 * 32 shape. This database consists low-resolution images (32 * 32), which allows researchers to experiment with new algorithms and neural networks.

Keywords-- CNN, Deep Learning, CIFAR-10, TensorFlow, Keras, machine learning, support vector machines, convolutional neural network, computer vision, neural networks, convolution, image classification.

I. INTRODUCTION:

Image classification is the process of classifying and labelling groupings of pixels or vectors inside an image based on specified rules, which is accomplished by applying computer vision and machine learning algorithms to extract information from the image. At its core, image classification is the task of assigning a label to an image based on a preset set of categories. And engineered two image classification methods: a deep learning neural network (CNN) [1] and a machine learning approach called support vector machines. The primary goal of a support vector machine (SVM) [2] is to identify the ideal hyperplane that optimally separates data points into distinct classes in a given feature space. The goal of SVM is to maximize the margin between the nearest data points (known as support vectors) from each class and the hyperplane. The ability of a convolutional neural network (CNN) to learn features automatically from raw pixel data. CNN is a type of artificial neural network that specializes in detecting and making sense of patterns, which makes it ideal for image analysis. In Shot CNN explains how it applies filters (kernels) to input images, computes dot products, extracts low-position and hierarchical features from input data, and uses RELU (rectified linear unit) activation to learn complicated patterns. The pooling subcaste for features maps to summarizing regions (max-pooling) to minimize feature dimensions. The fully connected layer converts the feature maps into vectors and connects them to output neurons for class prediction. For optimization and compile the model with a learning rate of 0.001, ADAM optimizer is utilized for optimization, and the loss function is sparse categorical cross entropy. The CNN model required millions of datasets for deep learning, which increased the dataset size through data augmentation [3]. The algorithms and neural networks were chosen in part due to their widely spread applications and proven success across multiple fields. The models would be tested against the dataset and compared based on their accuracy. The model's goal is to identify which is support vector machine (SVMs) [4] or convolutional neural networks (CNNs) [5] perform well and can be employed to perform image classification.

II. PROBLEM STATEMENT:

Develop a Convolutional Neural Network (CNN) model that can accurately classify images from the CIFAR-10 dataset into one of ten distinct categories which will create a classifier that can predict the correct class for each image in the test set, achieving high classification accuracy.

III. LITERATURE SURVEY:

Image classification is the most important phase in the image classification process. The image classification method may generate either a final or intermediate output. A variety of image classification techniques have been presented to date. Several studies have been undertaken to draw conclusions concerning the picture classification technique. Because of results and accuracy, it can be difficult to determine which technique is the best. Over the previous few decades, conventional approaches have been constantly modified, as have new image classification techniques, in order to obtain the most accurate results. Each classification technique has specific benefits and drawbacks. The two types of algorithms or neural networks are used. CNN (convolutional neural network) [1], which is a type of deep neural network, and SVM (support vector machine) [2], which is a type of machine learning algorithm. Both, that is, the neural network and algorithm, showed that the combination of both classifiers produced better results. The Support Vector Machine (SVM) is with kernels like linear, radial basis function (RBF) and polynomial kernels and with the penalty parameter, C, of the error term varying over a range of 0.0001 to 100. The SVM model with three kernels has underfit (data that is good enough but does not generalize to new data) and model performance. Because the training set has higher accuracy and the test set has lower accuracy, the testing set is not accurately testing the images in the model, resulting in the model overfitting the three kernels of the SVM model. In the SVM model with PCA (principal component analysis) [6], these number of components is 150 from 500. The test accuracy decreases from 40.3% to 39.2% if we decrease the number of components from 150 to 500. Hence, in addition to reducing training time, PCA also filters out noise in this scenario. In compared to the situation without dimensionality reduction, PCA increases accuracy on the test set from 30.6% (3072 dimensions) to 40.2% (150 dimensions). However, notice that the optimal kernel [7] in this case is the polynomial kernel in contrast to the linear kernel used earlier without dimensionality reduction. This is because the 150 components are in a different coordinate system that is better modelled by the polynomial kernel. The convolutional neural network (CNN) is a type of deep neural network. The model with accuracy of CNN model with training set accuracy is 93.6% and test set accuracy is 85.7%. The loss value of the testing set is 42% and the training set is 18%. The CNN model's accuracy is good, and the performance of the model is pretty amazing. The technique used in deep learning to artificially increase the size and variability of a training dataset by making small changes to existing data is called data augmentation and the model with the accuracy of the training set is 87% and the testing set is 85%. In the data augmentation model, the training and testing have good performance and improve generalization. Also mentioned the methodology and challenges in Table:1 Literature Survey.

Sr.no	author	methodology	Limitations/challenges
1	[1]	CNN (convolutional neural network).	improve generalization, pretty amazing (there is no more gap between train set and test set accuracy).
2	[2]	SVM (support vector machine).	Underfit (high accuracy of train set and low accuracy of test set).
3	[3]	Data Augmentation	improve generalization.
4	[6]	PCA (principal components analysis).	Test Accuracy decrease from 40.3% to 39.2% due to decrease the number of components from 150 to 500.
5	[5]	Optimization of CNN	ADAM is optimizer use for optimize the output that minimize and maximize model.
6	[7]	Performance of Evaluation of kernels in SVM model	Kernels are types of linear, polynomial, and RBF with different accuracy of training set and test set of kernels.
7	[10]	ANN (Artificial neural networks)	ANN having good performance and accuracy than SVM model.

Table 1. Literature Survey.

IV. SYSTEM ARCHITECTURE:

A. CNN MODEL:

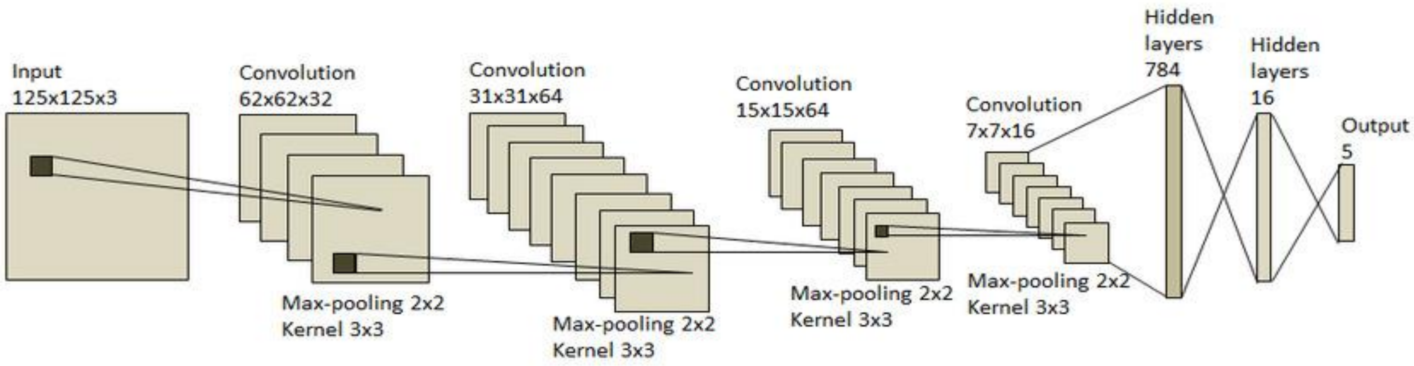

Fig .1 CNN Model

Fig. 2 CNN Architecture

B. SVM MODEL:

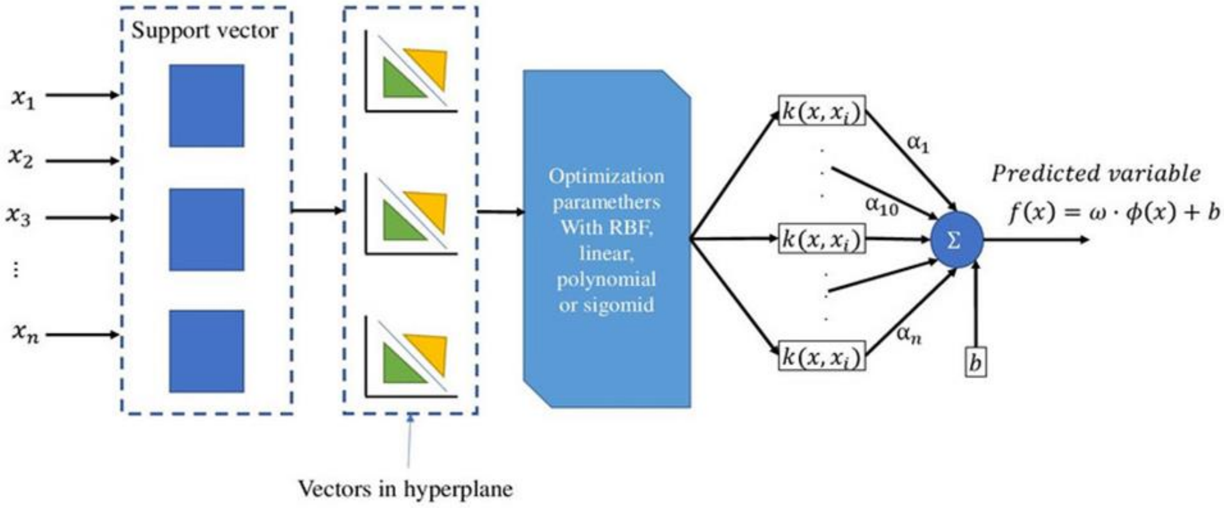

Fig. 3 SVM Model

V. PROPOSED METHODOLOGY:

➢ **CNN MODEL:**

1. Overview of Convolutional Neural Networks (CNN):

A Convolutional Neural Network (CNN) is a Deep Learning neural network design widely utilized in computer vision. Computer vision is a branch of artificial intelligence that allows computers to recognize and interpret images or visual data. Convolutional Neural Network (CNN) is an extension of artificial neural networks (ANN) [6] that is mostly used to extract features from grid-like matrix datasets. Convolutional neural networks (CNNs) [7] are deep neural networks that process grid-like data, such as pictures. Unlike standard neural networks, CNNs automatically learn spatial feature hierarchies using convolutional layers. They are made up of numerous layers, each of which has a specific role in processing and altering input data to extract significant patterns.

A convolutional neural network contains of multiple layers like the input layer, Convolutional layer, Pooling layer, and fully connected layers.

2. Convolutional Operations and Working:

- Input layer**:** A CNN's input layer [8] achieves the image's raw pixel values. A color image usually has three channels (RGB), whereas a grayscale image has only one. A color image with 32×32 pixels would have an input dimension of 32x32x3.

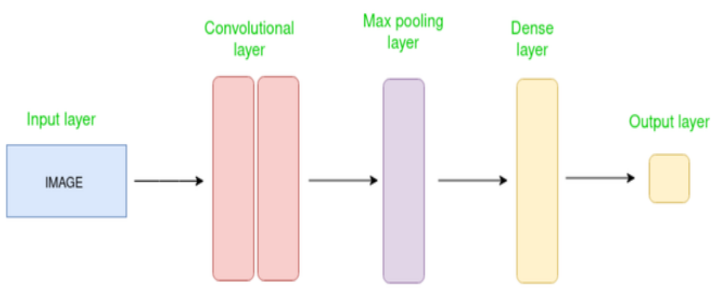

Fig. 4 CNN Architecture

- Convolutional Layers**:** The main operation of convolutional layers is convolution, which includes applying filters (kernels) to the input data. A filter is a small matrix (3x3) that moves across an input image and performs element-wise multiplication and summation to generate a single output value. Convolution is the process that produces a feature map or activation map. When a 3×3 filter is applied to a 5×5 input image, it slides over and calculates the dot product. The output feature map captures specific features like edges, corners, or textures based on the filter's learnt values.

3. Stride and Padding:

- Why padding is used?

Let us look at an example to better understand the requirement for padding. Consider a 5x5 input image with a 3x3 filter. When we use this filter in a convolutional operation on the input image, we receive a 3x3 feature map.

- What is padding?

Padding is a technique [9] for preserving the spatial dimensions of an input image following convolution operations on a feature map. Padding is the process of adding extra pixels around the boundary of the input feature map before convolution. Padding can help to reduce information loss at the input feature map's borders while also improving model performance.

- For a n*n input image and a f*f filter, the output feature map without padding has the shape (n-f+1) * (n-f+1).

- To maintain the spatial dimensions the same after convolution, we set n = (n-f+1), which refers to padding the input to (n+f-1) * (n+f-1). Where n*n is the dimension of the input picture, and f*f is the kernel or filter size.

- For example, the 5x5 image is padded to 7x7 by introducing a single pixel row/column boundary with zero values. This is known as zero padding.

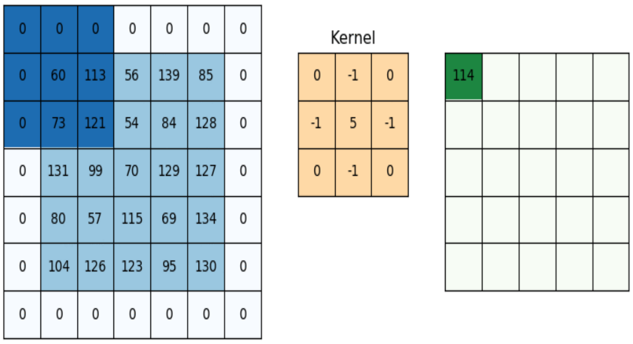

Fig. 5 Padding Operation

The formula for calculating output shape after padding is:

The output shape is (n + 2p — f + 1) * (n + 2p — f + 1). Where n is the input size, f is the filter size, and p is the padding value.

Strides: In convolution operations, the stride indicates how far the filter shifts across the input image with each application. By default, the stride is (1, 1), indicating that the filter shifts one pixel at a time. You can increase the stride value to make the filter skip pixels, resulting in a reduced output spatial dimension. The output shape formula for strides is:

You can increase the stride value to make the filter skip pixels, resulting in a reduced output spatial dimension. The output shape formula for strides is:

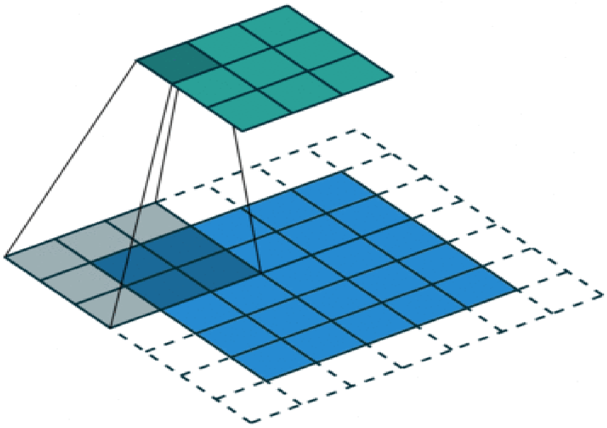

Fig. 6 Stride Operation

Output shape = ((n + 2p - f) / s + 1) * ((n + 2p - f) / s + 1), where s represents the stride value.

Higher strides make:

1. capturing higher-level features while opposing lower-level details.

2. Reducing computation requirements.

4. Activation Function:

Activation Functions are crucial in neural networks as they introduce nonlinearity, enabling the network to learn intricate patterns and descriptions. Without nonlinear activation functions, a neural network operates like a linear model, regardless of depth.

1. **The RELU activation function** is commonly employed in CNNs due to its simplicity and effectiveness. RELU(x) is defined as the maximum value between (0, x).

   This means that if the input value is positive, RELU returns the value directly; otherwise, it returns zero. RELU addresses the vanishing gradient problem, allowing for faster and more effective deep network training by keeping the gradient flow active and non-zero for positive inputs.

2. **The Sigmoid activation function** is commonly used in binary classification task output layers. It turns any input value into a 0–1 value that may be used in calculating probability. The formula for the sigmoid function is σ(x)=1/1+e^-x.

   While the sigmoid function is beneficial in some applications, it has vanishing gradients and converges slower than RELU.

3. **The Softmax activation function** converts the neural network's raw outputs into a probability distribution across input classes. Consider a multiclass classification task involving N classes. The Softmax activation produces an output vector of N entries, with the entry at index I representing the possibility that a specific input belongs to the class.

4. **Flattening and Output layer:** when the convolution and pooling layers, the resulting feature maps are flattened into a one-dimensional vector, which is then transmitted to a totally connected layer for categorizing. It takes previous layer's input and computes the final categorization output layer.

➢ **SVM MODEL:**

1. Overview of Support vector machine (SVM):

Support Vector Machine (SVM) [10] is a simple Supervised Machine Learning Algorithm that can be used for classification or regression. It is better suited to classification, although it can also be highly effective for regression. SVM basically finds a hyperplane that serves as the boundary between data kinds. In two dimensions, this hyperplane is simply a line. SVM plots each data item in the dataset in an N-dimensional space, where N represents the number of features/attributes in the data. Next, determine the best hyperplane to segregate the data. The dimension of the hyperplane is determined by the number of features. If the number of input features is two, the hyperplane is simply a line. Suppose. If there are three input features, the hyperplane becomes a two-dimensional plane. It's impossible to imagine when the number of features exceeds three. It can be used for calculating predictions about fresh data points by determining which side of the hyperplane they are on. Data points on one side of the hyperplane are allocated to one class, while data points on the opposite side are assigned to another class.

It can be used for calculating predictions about fresh data points by determining which side of the hyperplane they are on. Data points on one side of the hyperplane are allocated to one class, while data points on the opposite side are assigned to another class.

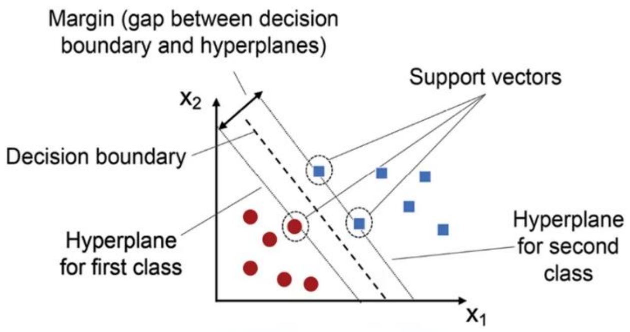

Fig. 6 Stride Operation

2. SVM Operations and Working:

- Define the Hyperplane: For a given set of labeled training data, the SVM searches a hyperplane that best separates the classes. The task is to find a linear decision boundary with the following equation: W^Tx+b=0, where W is the weight vector, X is the input feature vector, b is the bias element, and T is the weight vector's transpose.
- Maximize the Margin: The distance between a data point (x_i) and the decision boundary can be determined. The objective is to identify a hyperplane that maximizes the margin, which is defined as:
- Di = W^Tx+b / ||w||, where ||w|| denotes the Euclidean norm of the weight vector w. Euclidean norm of normal vector W.
- Linear SVM classifier: y for 1: W^Tx+b >= 0, y for 0: W^Tx+b <0.
- Optimization:
  The objective function to minimize is:

$$\frac{1}{2}||\mathbf{w}||^2 + C\sum_{i=1}^{N}\xi_i$$

  - C is a regularization parameter that determines the trade-off between maximizing margins and minimizing classification errors.
  - N is number of training samples.

- Popular kernel functions in SVM:

The SVM kernel converts low-dimensional input space into higher-dimensional space, converting non-separable issues to separable ones. It is mainly beneficial for nonlinear separation problems. They implemented a linear, polynomial, and RBF kernel [11].

- Linear: k (w, b) = $w^T x + b$
- Polynomial: k (w, x) = $(\gamma w^T x + b)^N$
- RBF: k (w, x) = $\exp(-\gamma||x_i - x_j||^n$

VI. RESUL AND DISCUSSION:

➢ SVM MODEL:

The Support vector machine (SVM) first set of classification experiments, the SVM model was tested on various kernels, with different regularization constants. The CIFAR-10 training dataset was imported from the keras library and had a shape of (49000, 32, 32, 3). The 32 x 32 x 3 color images were then flattened into a vector, reshaping the training dataset to have the shape (49000, 3072). Hence, it is important to note that only 49 thousand images were used when working on the SVM. However, since the training times were unfeasible on the entire set, the problem was more centered on the model itself than the availability of data.

The SVM Kernels and Regularization:

Three types of kernels [12] were used to train the data, namely the linear kernel, the radial basis function (RBF) kernel and the polynomial kernel. During the initial prototyping phase, a relatively small training set of 3000 images was used to train the data, and the penalty parameter, C determined using a separate validation set. In this way, one could relatively quickly determine the highest performing hyperparameter and respective kernel. The penalty parameter, C, of the error term was varied over the range 0.0001 - 100. The results are summarized in Table 1, Table 2, Table 3.

C	Training Accuracy (%)	Test Accuracy (%)
0.0001	35.4	98.0
0.001	48.4	18.8
0.01	70.9	25.3
0.1	98.9	28.6
1	100	27.9
10	100	27.9
100	100	27.9

Table 2. Linear Kernel

C	Training Accuracy (%)	Test Accuracy (%)
0.0001	10.7	7.9
0.001	10.7	7.9
0.01	10.7	7.9
0.1	30.7	11.9
1	48.7	11.9
10	84.8	10.5
100	100	11.9

Table 3. RBF Kernel

C	Training Accuracy (%)	Test Accuracy (%)
0.0001	10.7	8.7
0.001	10.7	8.7
0.01	12.0	8.7
0.1	27.2	12.5
1	72.2	26.5
10	96.3	25.6
100	99.8	24.7

Table 4. Polynomial Kernel

In Table 2, the best test accuracy of 28.6% was obtained for C=0.1. Following closely behind, the Polynomial Kernel demonstrated peak performance at an accuracy of 26.5%, with a penalty parameter C = 1. The plot of accuracy against C for the linear kernel is shown in Figure 1. The performance of the SVM on both the test set and training set improve substantially until C ≈ 0.1, after which the performance diminishes.

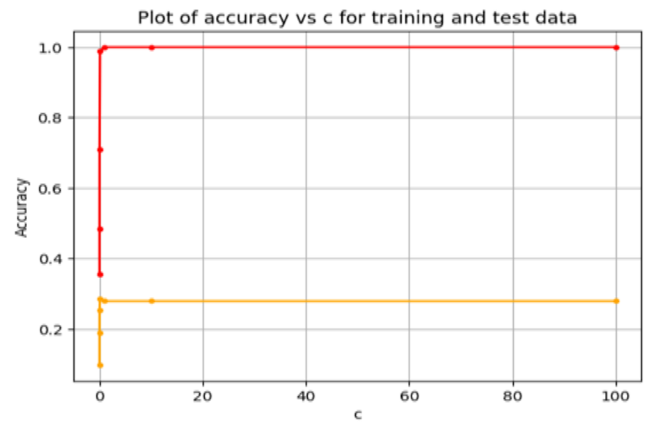

Fig. 8 Linear kernel

In Table 3, The RBF kernel was tested next, with the penalty parameter, C, ranging from 0.0001 to 100. In this case, the best test accuracy of 11.9% was obtained for C = 0.1. Note that after C = 0.1, although the training accuracy keeps increasing, the test accuracy starts to decrease, illustrating the effects of overfitting. The results are plotted in Figure 9.

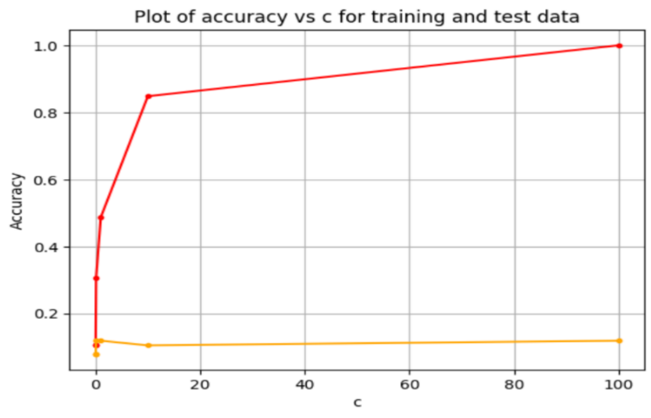

Fig. 9 RBF kernel

Thereafter, the Polynomial kernel was tested, with the penalty parameter, C, ranging from 0.0001 to 1000. In this case, the best test accuracy of 26.5% was obtained for C = 1. The results are plotted in Fig. 10.

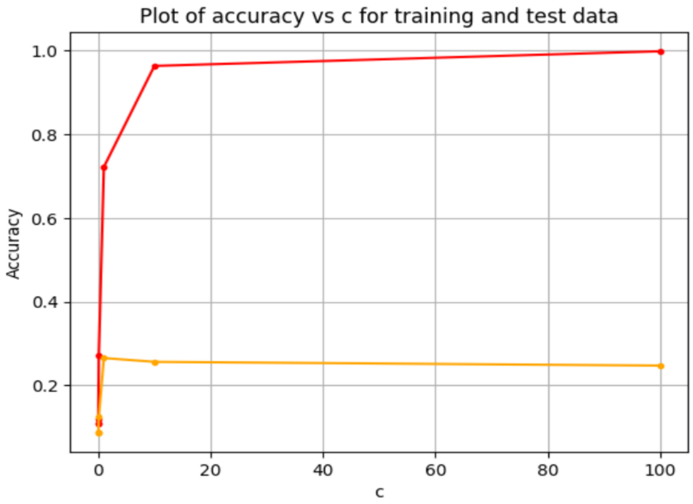

Fig. 10 Polynomial kernel

The Principal components anlaysis(PCA) [13] components that reduce the number of components to 150 and 500 Dimensions.

NO.OF.DIMENSIONS	Training Set %	Test Set %
150	67%	40.3%
500	71.1%	39.2%

Table 5. PCA Components

Therefore, we see that the test accuracy decreases from 40.3% to 39.2% if we decreases the number of components from 150 to 500. Hence, the addition to decreasing the time of train the dataset, PCA also serves to filter noise in the case.

- **CNN MODEL:**

The Convolutional Neural Network (CNN) is used to train and test the model. It employs a convolution layer, the Conv2d layer, as well as pooling and normalization approaches. Finally, it comes into a dense layer followed by a final dense layer, which serves as our output layer. We're implementing a RELU activation function, and the output layer's function is SOFTMAX. Lets After building the CNN model, compile it using the optimizer 'ADAM' [14], loss'sparse categorical crossentropy', and metrics 'accuracy'. Let's fit the model with x_train and y_train, then validate with x_test and y_test. Train the model for 30 epochs. Initially, as the number of epochs increases, the model learns more from the training data, and prediction accuracy improves across both the training and validation datasets. This is because the model has more possibilities for manipulating its weights and biases in order to minimize the loss function. Data augmentation is a deep learning technique that artificially increases the quantity and variability of a training dataset by making minor changes to existing data. It's a common technique in deep learning since it can help increase model performance while preventing overfitting. To expand the dataset's size, photos were generated using the image data augmentation technique [15]. Let us fit the model and generate an image. Train the model for 30 epochs.

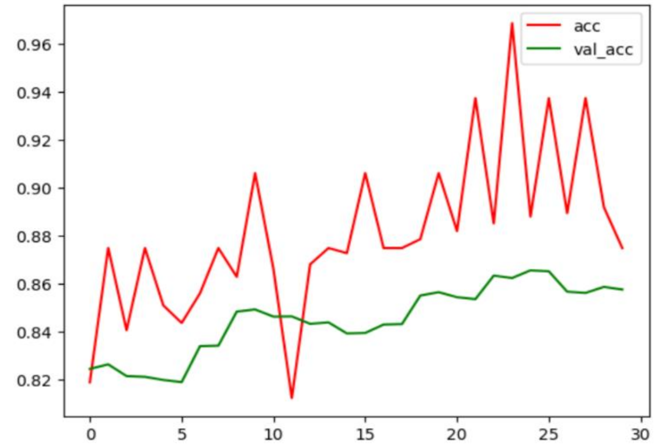

Fig. 11. Accuracy and val_accurcy

Lets evaluate the model for training and testing set. The accuracy of training is 93.6% and testing accuracy is 85.7%. The loss value of training set is 43% and testing set is 18.1%.

	Training set	Test set
Accuracy	93.6%	85.7%
Loss	43%	18.1%

Table 7. Accuracy and Loss

VII. LIMITATIONS:

- One major issue in image classification research is the distribution of classes being imbalanced.
- Another issue is a lack of labeled data, which hinders classification.
- Additionally, the complex complexity of visual information makes effective categorization and labeling challenging.
- Traditional image classification approaches, like feature extraction and pattern recognition, have limitations in accuracy and time consumption.
- It takes more time in training and data augmentation. Each epochs take 5 min to training.
- Furthermore, the number of images and classes in large-scale image classification begins challenges to evaluating similarity efficiently and training supervised classification models for newly developing classes

- Image classification algorithms require largely on training data quality and quantity to work effectively. The research does not explore how variations in data can impact the accuracy of the classifiers. This is a fundamental part of machine learning.
- The process of tuning hyperparameters and optimizing the model can be intricate and time-consuming.

- Using a pre-built deep convolutional neural network may be difficult as it required significant computing power and technical knowledge. This can make it difficult for persons without significant expertise or powerful hardware to use these models effectively.

VIII. CONCLUSION AND FUTURE SCOPE:

- Conclusion:

In this paper, we used Convolutional Neural Networks (CNN) and Support vector machine (SVM) for image classification using image dataset that is CIFAR-10 dataset. The CNN are important in the field of computer vision and have a wide range of applications. The CNN model performance for image classification gives an accuracy value of 93% in training set and 85% in testing set. The increase size of dataset by help of data augmentation in CNN model. The SVM model Performance for image classification gives an accuracy value of 100% in training set and 28% in testing set. Therefore, the SVM model gets underfitted due to high accuracy of training set and low accuracy of testing set. The performance of the CNN model is good and improve generalization, pretty amazing. The SVM not performing well due to underfitted of model. So, the CNN model is pretty amazing for image classification. The CNN gives accurate result of image classification with labeled and accuracy.

- Future Scope:

- In future the image classification by using VIT that is vision Transformer to accurate prediction of images classification on Image Net dataset with 1000 of images categories and VIT gives very good and amazing performance and accuracy.
- Image classification by using famous pre-trained models that is VGG16, VGG19, Resnet, Alexnet and Mobile Net etc.
- By using transfer learning technique for advanced image classification.
- Automated Machine Learning (AutoML) techniques can be tested on CIFAR-10 to optimize architectures, hyperparameters, and augmentations, which are then scaled to larger tasks.

REFERENCES:

1. O'Shea, Keiron & Nash, Ryan. (2015). An Introduction to Convolutional Neural Networks. ArXiv e-prints [1].

2. Evgeniou, Theodoros & Pontil, Massimiliano. (2001). Support Vector Machines: Theory and Applications. 2049. 249-257. 10.1007/3-540-44673-7_12 [2].

3. Shorten, Connor & Khoshgoftaar, Taghi. (2019). A survey on Image Data Augmentation for Deep Learning. Journal of Big Data. 6. 10.1186/s40537-019-0197-0 [3].

4. O'Shea, Keiron & Nash, Ryan. (2015). An Introduction to Convolutional Neural Networks. ArXiv e-prints [4].

5. Howley, Tom & Madden, Michael & O'Connell, Marie-Louise & Ryder, Alan. (2005). The Effect of Principal Component Analysis on Machine Learning Accuracy with High Dimensional Spectral Data. 209-222. 10.1007/1-84628-224-1_16 [5].

6. Grossi, Enzo & Buscema, Massimo. (2008). Introduction to artificial neural networks. European journal of gastroenterology & hepatology. 19. 1046-54. 10.1097/MEG.0b013e3282f198a0 [6].

7. Purwono, Purwono & Ma'arif, Alfian & Rahmaniar, Wahyu & Imam, Haris & Fathurrahman, Haris Imam Karim & Frisky, Aufaclav & Haq, Qazi Mazhar Ul. (2023). Understanding of Convolutional Neural Network (CNN): A Review. 2. 739-748. 10.31763/ijrcs.v2i4.888 [7].

8. Wang, Zhichen & Li, Hongliang. (2020). Research on a convolution operation method based on domain transformation in deep learning. Journal of Physics: Conference Series. 1550. 032132. 10.1088/1742-6596/1550/3/032132 [8].

9. Alantali, Fatmah & Halawani, Yasmin & Mohammad, Baker & Al-Qutayri, Mahmoud. (2021). SLID: Exploiting Spatial Locality in Input Data as a Computational Reuse Method for Efficient CNN. IEEE Access. PP. 1-1. 10.1109/ACCESS.2021.3071409 [9].

10. Srivastava, Durgesh & Bhambhu, Lekha. (2010). Data classification using support vector machine. Journal of Theoretical and Applied Information Technology. 12. 1-7 [10].

11. Shadeed, Intisar & Abd, Dhafar & Alwan, Jwan & Rabash, Abubaker. (2018). Performance Evaluation of Kernels in Support Vector Machine. 96-101. 10.1109/AiCIS.2018.00029 [11].

12. Schölkopf, Bernhard & Smola, Alexander. (2018). Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond. 10.7551/mitpress/4175.001.0001 [12].

13. Almaiah, Drmohammed & Almomani, Omar & Saaidah, Adeeb & Alotaibi, Shaha & Bani-Hani, Nabeel & Al Hwaitat, Ahmad & Al-Zahrani, Ali & Lutfi, Abdalwali & Awad, Ali & Aldhyani, Theyazn. (2022). Performance Investigation of Principal Component Analysis for Intrusion Detection System Using Different Support Vector Machine Kernels. Electronics. 11. 10.3390/electronics11213571 [13].

14. M. Dhouibi, A. K. Ben Salem and S. Ben Saoud, "Optimization of CNN model for image classification," 2021 IEEE International Conference on Design & Test of Integrated Micro & Nano-Systems (DTS), Sfax, Tunisia, 2021, pp. 1-6, doi: 10.1109/DTS52014.2021.9497988. keywords: {Computer vision;Computational modeling;Neurons;Computer architecture;Throughput;Topology;Complexity theory;CNN;Topology;Accuracy;Pruning;Parameters;FPGA;Image classification}, [14].

15. A. Mikołajczyk and M. Grochowski, "Data augmentation for improving deep learning in image classification problem," 2018 International Interdisciplinary PhD Workshop (IIPhDW), Świnoujście, Poland, 2018, pp. 117-122, doi: 10.1109/IIPHDW.2018.8388338. keywords: {Image color analysis;Machine learning;Lesions;Image classification;Neural networks;Cancer;Task analysis;Machine learning;style transfer;data augmentation;deep learning;medical imaging}, [15].