Scientific Journal of Impact Factor (SJIF): 7.37
E-ISSN (O): 2348-4470
P-ISSN (P): 2348-6406

International Journal of Advance Engineering and Research Development

Volume 13, Issue 6, June 2026

IMAGE CLASSIFICATION USING ARTIFICIAL NEURAL NETWORKS

Chetan Hirjibhai Vanapariya[1]

[1]*Computer Engineering Department, Takshshila College of Engineering & Technology, Rajkot, India*

Abstract — *An Artificial Neural Network is basically a computing system that imitates the functionality and structure of a human brain. Just like the brain uses neurons to process data and make decisions, Artificial Neural Networks use artificial neurons to analyze data, identify patterns and make predictions. These networks consist of layers of interconnected neurons that work together to solve complex problems. Now these layers learn, adapt and solve tough problems in sectors like image classification, speech processing, natural language processing and weather prediction, making them a key part of new technology trends. An Artificial Neural Network is a computational model inspired by the human brain. It consists of interconnected processing units called neurons that learn to recognize patterns, process complex data, and make predictions. Artificial Neural Networks serve as the backbone for modern machine learning and artificial intelligence. These networks have input, hidden and output layers. Image classification is the process of assigning a predefined label to an image based on its visual content. The goal is to enable a model to automatically recognize patterns, textures and shapes to categorize images into classes it has learned during training correctly. In deep learning, image classification is a common task where a model learns to recognize objects, scenes, or patterns in images. A trained model can classify images into categories like cat, dog or elephant based on features it has extracted from the input images.*

Keywords *- Artificial Neural Networks; Image Classification; ANN; CNN; VGG; Image Processing; Deep learning; Machine Learning;*

I. INTRODUCTION

Artificial Neural Networks (ANNs) are a subset of machine learning models that are composed of interconnected neurons, structured to simulate the way the human brain processes information. An Artificial Neural Network (ANN) is a computational model inspired by the structure and functioning of the human brain. Neural networks have emerged as a key component in modern artificial intelligence. ANN stands at the forefront of innovation in an era of rapidly evolving technology.

The main idea is how our brain learns from past experience and makes decisions, similar way ANNs can learn from the available data as they process. Artificial Neural Networks have superior learning abilities. ANN is a multi-layer network build by collection of interconnected nodes called neuron. The neural network is also called as a perceptron. Each neuron is responsible for receiving input, processing it, and transmitting output to other neurons in the network.

Artificial Neural Networks (ANNs) have been widespread usage in a numerous amount of application domains. The most of applications work on feed forward ANNs and the back propagation. Computers have difficulties while recognizing simple patterns much less generalizing those patterns of the past into actions of the future. Now, innovations in biological research provide an initial understanding of the natural thinking process. This research shows that brains store information as patterns. Some of these patterns are very complicated and allow us the ability to recognize human faces from many different angles. This process of storing information as patterns, utilizing those patterns, and then solving problems generating a new field in computing. ANNs are used in a wide variety of machine learning and deep learning applications such as image classification, natural language processing, time series prediction, computer vision, natural language processing, sentiment analysis in text, self-driving cars, healthcare and robotics.

The key components of Artificial Neural Networks as below.

A. Neural Network layers.
B. Activation Functions.
C. Optimization Algorithms.

A. The architecture of an Artificial Neural Networks (ANNs) is typically divided into three primary layers as below.

Input Layer: This layer receives the initial data or input features and transmits them to the subsequent layers for processing. For example, in an image classification task, the input could be an image.

Hidden Layers: These intermediate layers perform complex computations on the input data, extracting meaningful features and patterns. These layers process the data received from the input layer. The more hidden layers there are, the more complex patterns the network can learn and understand.

Output Layer: This is where the final decision or prediction is made. For example, after processing an image, the output layer might classify whether it is a cat or a dog.

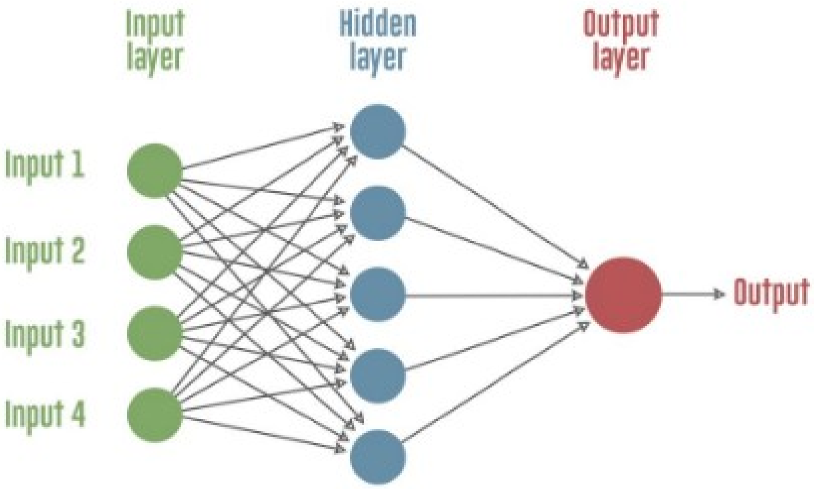

Figure 1. Architecture of Artificial Neural Networks (ANNs)

B. Activation functions in ANNs.

Activation functions are important in neural networks because they introduce non-linearity and helps the network to learn complex patterns. Lets see some common activation functions used in ANNs:

Softmax: Converts raw outputs into probabilities used in the final layer of a network for multi-class classification tasks.

Sigmoid Function: Outputs values between 0 and 1, making it suitable for binary classification.

ReLU (Rectified Linear Unit): Outputs the input directly if positive, otherwise zero. It is widely used due to its efficiency. It helps to solve the vanishing gradient problem.

Tanh (Hyperbolic Tangent): Similar to the sigmoid but outputs values between -1 and 1, which can help with training in certain situations. It is a scaled version of the sigmoid.

Leaky ReLU: A variant of relu that allows small negative values for inputs helps in preventing dead neurons during training.

C. Optimization Algorithms in ANNs.

Optimization algorithms adjust the weights of a neural network during training to minimize errors. The goal is to make the network’s predictions more accurate. The key algorithms are as below:

Gradient Descent: Most basic optimization algorithm that updates weights by calculating the gradient of the loss function.

Stochastic Gradient Descent (SGD): Updates weights using a single training example per iteration.

Adam (Adaptive Moment Estimation): Combines momentum and adaptive learning rates for more efficient optimization.

RMSprop: A variation of gradient descent that adjusts the learning rate based on the average of recent gradients, it is useful in training recurrent neural networks (RNNs).

II. CLASSIFICATION USING ARTIFICIAL NEURAL NETWORKS

The process of image classification in ANNs.

Artificial Neural Networks (ANNs) learn by adjusting the weights of connections between neurons based on the error between the prediction and actual outputs. ANNs are trained on labeled data. The network is given inputs along with the correct outputs, and it trains by adjusting weights to reduce the difference between the prediction and actual outputs. The main method for this is back-propagation, where the errors are calculated at the output layer and moved backward through the network to change the weights. This process continues until the ANN reaches at expected level of accuracy. The training process of an artificial neural network is an iterative process in which the calculations are carried out forward and backward through each layer in the network until the error function is minimized.

In artificial neural networks each neuron in a hidden layer receives the signals from all of the neurons in a layer above it, typically an input layer. After a neuron performs its function it passes its output to all of the neurons in the layer below it, providing a feed-forward path to the output. In an ANN, each artificial neuron receives one or more inputs and processes them using an activation function, which determines the neuron's output. The output of each neuron is transmitted to other neurons through weighted connections, which are used to adjust the contribution of each neuron to the final output. It uses back-propagation and gradient descent to minimize errors. The process of minimizing the difference between the desired output and the actual output is carried out using optimization algorithms, such as gradient descent, stochastic gradient descent, Adam. ANN requires labeled data for training. Their tasks are like image classification and predicting output.

Multilayered feed forward neural network based classifier is used for classification. A multi-layer feed forward neural network consists of a layer of input units, one or more layers of hidden units, and one layer of output units. The output from each layer is the weighted total of all input vectors along with the bias term, passed through some activation function. The network weight adjustment is done by back-propagating the error of the network. The learning algorithms used for learning weights in the network are gradient descent with Momentum.

For example, to teach an ANN to recognize a cat, we feed ANN to thousands of images of cats. The network processes these images and learns to identify the features of a cat. When the network has been trained, we test it by providing new images to see if it can correctly identify cats. The network's prediction is then compared to the actual result. If it makes a false prediction, the network adjusts by fine-tuning the weights of the connections between neurons using a process called back-propagation. This makes correcting the weights based on the difference between the predicted and actual output. This process iterates until the network can accurately recognize a cat in an image with minimum error. So ANN can classify images with high accuracy and less error.

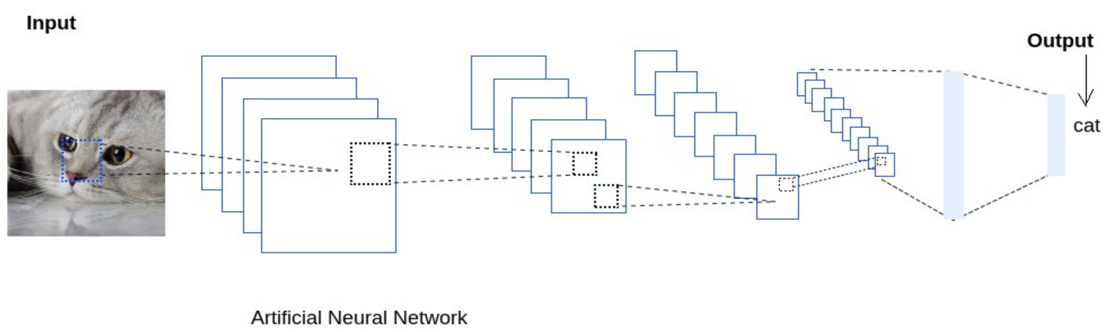

Figure 2. Input Image and Output as Classification of ANN

III. TRAINING AND TESTING OF ARTIFICIAL NEURAL NETWORKS

A training set is presented to the model. The training set constitutes of input examples and corresponding target outputs. The inputs are noted for the response of the network, and the weights between with networks are adjusted for error reduction, for the attainment of the desired output. The network follows successive iterations during this process until the computed result converges to the correct one. Construction of the training set requires special consideration. A training set is considered an ideal one, and it should be giving a better representation of the underlying model. Otherwise, a reliable model with desirable results cannot be achieved with an unrepresentative training set.

During training, the network is shown examples like images of cats and learns to recognize patterns in them. After training, the network is tested on test data to check its performance. The better the network is trained, the more accurately it will predict new data.

There are three types of sets in which sample data is distributed: (1) the training set, (2) the validation set, and (3) the testing set. The training set is used to train the ANN model; it is a set of sample data that is used to modify or adjust the weights in the ANN to produce the desired outcome. The validation set is used to inform the ANN when training is to be terminated. The test set provides an entirely independent way of examining the precision of the ANN. The test set is a set of sample data that is used for the evaluation of the ANN model.

Performance metrics in machine learning are used to evaluate the performance of a machine learning model. These metrics provide quantitative measures to assess how well a model is performing and to compare the performance of different models. Performance metrics are important because they help us understand how well our model is performing and whether it is meeting our requirements. In this way, we can make informed decisions about whether to use a particular model or not.

There are various metrics which we can use to evaluate the performance of ML algorithms, classification as well as regression algorithms. We are going to discuss various performance metrics that can be used to evaluate predictions for classification problems.

Confusion Matrix:

The confusion matrix is the easiest way to measure the performance of a classification problem where the output can be of two or more type of classes. A confusion matrix is nothing but a table with two dimensions viz. Actual and Predicted and furthermore, both the dimensions have True Positives (TP), True Negatives (TN), False Positives (FP), False Negatives (FN) as shown below.

True Positives (TP) − It is the case when both actual class & predicted class of data point is 1.
True Negatives (TN) − It is the case when both actual class & predicted class of data point is 0.
False Positives (FP) − It is the case when actual class of data point is 0 & predicted class of data point is 1.
False Negatives (FN) − It is the case when actual class of data point is 1 & predicted class of data point is 0.

Accuracy:

Accuracy is most common performance metric for classification algorithms. It may be defined as the number of correct predictions made as a ratio of all predictions made.

Classification Report:

This report consists of the scores of Precision, Recall, F1 and Support.

Precision:

Precision measures the proportion of true positive instances out of all predicted positive instances. It is calculated as the number of true positive instances divided by the sum of true positive and false positive instances.

Recall:

Recall measures the proportion of true positive instances out of all actual positive instances. It is calculated as the number of true positive instances divided by the sum of true positive and false negative instances.

F1 Score:

F1 score is the harmonic mean of precision and recall. It is a balanced measure that takes into account both precision and recall. Mathematically, F1 score is the weighted average of the precision and recall. The best value of F1 would be 1 and worst would be 0.

ROC AUC Score:

The ROC (Receiver Operating Characteristic) Area Under the Curve(AUC) score is a measure of the ability of a classifier to distinguish between positive and negative instances. It is calculated by plotting the true positive rate against the false positive rate at different classification thresholds and calculating the area under the curve.

As name suggests, ROC is a probability curve and AUC measure the separability. In simple words, ROC-AUC score will tell us about the capability of model in distinguishing the classes. Higher the score, better the model.

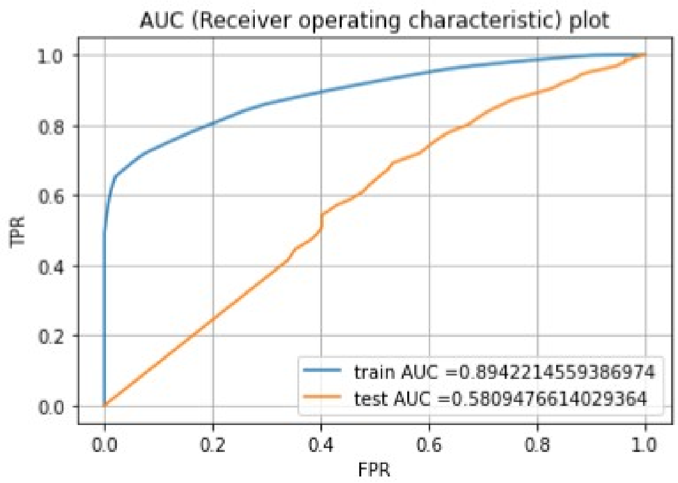

Figure 3. ROC Curve on Training and Testing Data

IV. CONCLUSION

Artificial Neural Networks are computing systems inspired by biological neural networks. Like the human brain, they learn by examples, supervised or unsupervised. Deep Neural Networks are ANNs with a larger number of layers. Usually, we can call a network deep if it has at least 2 hidden layers. In some cases, this threshold can go up to 10 layers. Some deep neural networks may have thousands of layers. Each layer transforms the input into more abstract representations. Training a neural network means feeding it lots of data. It learns step by step and improves each time. But building a good neural network needs the right number of layers and correct settings. If done well, ANN can give powerful results, just like a trained brain working on a task. From feedforward neural networks that perform basic classification tasks to advanced architectures like recurrent neural networks (RNNs) and convolutional neural networks (CNNs) that handle sequential and visual data, ANNs are at the forefront of technological innovation.

Training and testing deep learning models for image classification can be challenging, especially with large datasets. However, with careful hyper-parameter tuning and by applying strategies like data augmentation, transfer learning, and batch size optimization, you can significantly reduce training times and improve model accuracy. By continually experimenting with different hyper-parameter values and optimizing your model’s architecture, you’ll be able to achieve better performance without requiring enormous computational resources. This overview of image classification challenges and solutions provides a foundation for addressing model complexity and training and testing bottlenecks. With this knowledge, you can make your deep learning workflows more efficient and deploy models more effectively, even on large-scale datasets.

REFERENCES

[1] K., Zhang, X., Ren, S., & Sun, “Deep residual learning for image recognition”, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778, June 2016.

[2] Dai, Y., & Zhu, “A machine vision system for micro-milling tool condition monitoring”, Precision Engineering, pp. 183–191, 2018.

[3] Sivabalan, K. N., & Ghanadurai, “Detection of defects in digital texture images using segmentation”, International Journal of Engineering, Science and Technology, pp. 5187–5191, 2010.

[4] Scime, L., & Beuth, “Anomaly detection and classification in a laser powder bed additive manufacturing process using a trained computer vision algorithm”, Additive Manufacturing, pp. 114–126, 2018.

[5] Zhang, Y., Hong, G. S., Ye, D., Fuh, J. Y. H., & Zhu, “Powder-bed fusion process monitoring by machine vision with hybrid convolutional neural networks”, IEEE Transactions on Industrial Informatics, pp. 5769–5779, 2020.

[6] Ansari, M. A., Crampton, A., Garrard, R., Cai, B., & Attallah, “A convolutional neural network (CNN) classification to identify the presence of pores in powder bed fusion images”, International Journal of Advanced Manufacturing Technology, pp. 5133–5150, 2022.

[7] Szeliski, R. , “Computer vision: Algorithms and applications”, Springer Science & Business Media, 2010.

[8] Nasrabadi, N. M., & King, R. A. , “Image coding using vector quantization: A review”, IEEE Transactions on Communications, pp. 957–971, 1988.

[9] S. Sharma and R. Gupta, "Performance Analysis of Python with OpenCV and MATLAB for Image Processing," in Proc. IEEE ICCMC, 2018.

[10] M. Zhang and L. Chen, "Python vs Java: Image Processing and Computer Vision Perspectives," J. Real-Time Image Process., vol. 18, 2021.

[11] P. Srinivas and K. Reddy, "A Performance Comparison of MATLAB and Python in Image Filtering Applications," IEEE Access, vol. 8, 2020.

[12] Krizhevsky, A., Sutskever, I., & Hinton, “ImageNet classification with deep convolutional neural networks”, Advances in Neural Information Processing Systems, G.E., pp. 1097–1105, 2012.

[13] V. Agarwal, "Image Processing with Python: An Alternative to MATLAB", Elsevier, 2019.

[14] Y. Tay, P. Lallican, M. Khalid, C. Viard-Gaudin, S. Knerr, “An Offline Cursive Handwriting Word Recognition System ”, Proc. IEEE Region 10 Conf., (2001).

[15] Sivamurugan, C., B. Perumal, Yelchuri Siddhartha, Vavilala Krishna Murthi, Yaramati Nanda Sankar, and Yerra Varun. "Enhanced Blood Group Prediction with Fingerprint Images using Deep Learning." In 2024 3rd International Conference on Automation, Computing and Renewable Systems (ICACRS), IEEE, pp. 1091-1097, 2024.

