iJRASET
International Journal For Research in Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 12 **Issue:** X **Month of publication:** October 2024

DOI: https://doi.org/10.22214/ijraset.2024.64911

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 12 Issue X Oct 2024- Available at www.ijraset.com

Advanced Transfer Learning Strategies for Improving Deep Learning Model Efficiency and Accuracy

N. Balakumar[1], R. Amsaveni[2], K. Brindha[3] S. Sasikala Devi[4], N. Manoharan[5], Mani Gopalsamy[6], L. Sivakami[7]
[1, 2, 4, 5]*Department of Computer Science,* [3]*Department of Data Science,* [6]*Department of Cyber Security,* [5]*Department of Mathematics and statistics*
[1]*United College of arts and sciences, Coimbatore*
[2]*Sree Saraswathi Thyagaraja College Pollachi,*
[3]*Sri Krishna Adithya College of arts and science*
[4]*Govt. Arts and Science College, Mettupalayam*
[5]*Jamal Mohamed College, Trichy*
[6]*Coolsoft LLC,* [7]*SRM Institute of Science and Technology, Kattankulathur*

Abstract: ***Machine learning primarily focuses on supervised learning, which relies on labeled data, while unsupervised learning addresses data without labels. Reinforcement learning, on the other hand, develops agents that learn optimal actions based on past experiences. Transfer learning, distinctively, enhances learning in a target task by leveraging knowledge from related source tasks. In recent years, many machine learning models have remained single-task focused. This paper explores scenarios in which transfer learning can be applied effectively from a source domain to a target domain, employing feature selection, extraction, and construction techniques. These methods enable transformation into a new feature space by generating a refined subset of the original features.***
Keywords: Deep Learning, Supervised Learning, Unsupervised Learning, Reinforcement Learning, And Transfer Learning.

I. INTRODUCTION

Deep learning, also known as deep structured or hierarchical learning, is a subset of machine learning that centers on representation learning as opposed to traditional task-specific algorithms. Deep learning models, which include deep neural networks (DNNs), deep belief networks (DBNs), and recurrent neural networks (RNNs), are pivotal in supervised, semi-supervised, and unsupervised learning approaches. These models have achieved remarkable success across a range of applications, including computer vision, speech recognition, natural language processing, audio recognition, social network filtering, machine translation, bioinformatics, and even drug discovery. In many of these fields, deep learning methods have either matched or surpassed human expertise in performance. The predominant architectures in deep learning encompass Deep Boltzmann Machines (DBMs), DBNs, Convolutional Neural Networks (CNNs), Artificial Neural Networks (ANNs), and RNNs. While inspired by biological neural systems, deep learning structures diverge considerably from actual neurobiological patterns, limiting direct applicability to neuroscience.

Transfer learning addresses an important limitation in machine learning by focusing on reusing knowledge gained in one task to enhance performance in a related but distinct task. For instance, knowledge of recognizing cars could aid in identifying trucks. Though connected conceptually to the transfer of learning in cognitive psychology, formal cross-disciplinary connections are sparse. The paper is structured as follows: Section II explores deep learning, with sub-sections on supervised and unsupervised learning and their key architectures, including CNNs, ANNs, and RNNs in supervised learning, and DBNs, Autoencoders, Generative Adversarial Networks (GANs), Self-Organizing Maps (SOMs), and DBMs in unsupervised learning. Section III delves into Transfer Learning, detailing its forms—Inductive, Transudative, and Unsupervised Transfer Learning. Section IV discusses Transfer Learning's characteristics, especially how it impacts base and target datasets across varying domains and data sizes. Section V concludes with insights on Transfer Learning's effects and potential future directions.

II. DEEP LEARNING

Introduced to machine learning by Rina Dechter in 1986 and to artificial neural networks by Igor Aizenberg in 2000, deep learning's foundation was laid with multilayer perceptrons for supervised tasks, initially demonstrated by Alexey Ivakhnenko and V.G. Lapa in 1965. Since then, deep learning has evolved, with Kunihiko Fukushima's 1980 Neocognitron being an early model for computer vision. In 1989, Yann LeCun's application of backpropagation in deep neural networks advanced handwritten ZIP code recognition, though training required extensive time [4].

©IJRASET: All Rights are Reserved

Transfer learning, first explored by Lorien Pratt in 1993 with the discriminability-based transfer (DBT) algorithm, later evolved in 1997 with multi-task learning theories, which were formalized in Learning to Learn by Pratt and Sebastian Thrun in 1998. The reuse of neural networks, an important application in cognitive science, was also recognized in a 1996 special issue of Connection Science. Our study benefitted from the research carried out on advanced breast cancer detection approaches using deep learning [35-37]

A. *Supervised Learning*

Supervised learning develops a function that maps inputs to outputs based on labeled training data [8]. The algorithm aims to generalize from examples to predict unseen data accurately. In supervised learning, core architectures include:

1) *Convolutional Neural Networks (CNNs):* CNNs, designed for minimal preprocessing, are commonly applied in image analysis. Their architecture leverages weight-sharing and translation invariance, with CNNs inspired by biological vision systems where each neuron responds within a specific visual field. This architecture enables CNNs to effectively process images with little human-engineered feature extraction, excelling in image recognition, recommendation systems, and NLP [11].
2) *Artificial Neural Networks (ANNs):* ANNs are inspired by biological neural networks and use interconnected artificial neurons. Each neuron processes inputs and transmits signals, while connections (edges) carry weight, which adjusts during learning. ANNs, originally developed to mimic human problem-solving, are now tailored for specific tasks, including visual and audio recognition, social network filtering, and medical diagnostics [12].
3) *Recurrent Neural Networks (RNNs):* RNNs have a feedback architecture that enables them to process sequential data, such as text or time series. This dynamic nature is especially valuable for tasks like handwriting and speech recognition. RNNs exhibit either finite or infinite impulse responses, the latter providing continuous feedback, as in LSTM networks for memory-based processes [13].

B. *Unsupervised Learning*

Unsupervised learning identifies patterns in unlabelled data, often for clustering or dimensionality reduction. Unlike supervised learning, it lacks straightforward accuracy metrics.

1) *Deep Belief Networks (DBNs):* DBNs, generative graphical models composed of layers of latent variables, can learn to reconstruct inputs probabilistically. They are often constructed using RBMs, where each layer's hidden units hierarchically detect features, making them suitable for applications such as drug discovery and bio-signal analysis [14].
2) *Autoencoders:* Autoencoders learn compressed data representations, typically for dimensionality reduction, with applications extending to generative modeling. Their bottleneck layer enforces minimal data representation, valuable in high-dimensional data tasks [15].
3) *Generative Adversarial Networks (GANs):* GANs use two neural networks in a zero-sum framework where one generates data samples while the other evaluates them. This approach has been highly successful in generating realistic images and audio [17].
4) *Self-Organizing Maps (SOMs):* SOMs reduce dimensionality by mapping high-dimensional data into low-dimensional grids, preserving topological properties. This approach is useful in data clustering and visualization tasks [18].
5) *Deep Boltzmann Machines (DBMs):* DBMs, energy-based models, can represent complex distributions and perform generative tasks through stochastic sampling. With constrained connectivity, DBMs achieve efficient training, applicable in areas like electroencephalography analysis [10].

III. TRANSFER LEARNING

Transfer learning optimizes training on a new task by utilizing knowledge from previously learned tasks, significantly reducing the computational cost and improving model accuracy and robustness [19, 23]. This is achieved by adapting parameters, features, or entire models learned from a source domain to a target domain, where acquiring large datasets for the target domain may be challenging. Given its ability to enhance generalization in new environments, transfer learning has become central in machine learning research and applications, particularly where data scarcity and time efficiency are crucial constraints.

A. *Inductive Transfer Learning*

In inductive transfer learning, the source and target tasks differ, but the model's underlying knowledge is transferable to help optimize performance on the new task. This approach is useful when labeled data in the target domain is available, enabling the model to better generalize to unseen scenarios in the target task by fine-tuning both general and task-specific features.

1) *Process:* The model is trained on the source dataset and then partially or entirely fine-tuned on the labeled target dataset. The transferability of layers decreases from early to later layers; thus, early layers often capture low-level, reusable features, whereas later layers learn task-specific details.
2) *Applications:* Commonly used in image classification, where models trained on large-scale datasets like ImageNet [27] are adapted to specific tasks like medical imaging or satellite image classification. This approach also finds utility in NLP, where large language models (e.g., BERT, GPT) are fine-tuned on task-specific datasets.

B. *Transductive Transfer Learning*

Transductive transfer learning is suitable when only unlabeled target data is available during source training. Here, the model uses domain similarity to improve performance, making this approach relevant in domains where labeling new data is expensive or infeasible. Transductive transfer focuses on learning domain-invariant representations that can generalize well across both source and target domains.

1) *Process:* During training, the model leverages unlabeled target data through domain adaptation techniques like adversarial learning or Maximum Mean Discrepancy (MMD), which minimize domain discrepancy and make the learned features robust to domain shifts.
2) *Applications:* Widely applied in semi-supervised and unsupervised domains, such as sentiment analysis, where labeled source data might come from one domain (e.g., movie reviews) and needs to be applied to a different domain (e.g., product reviews). Another common use case is in object recognition across different lighting or background environments.

C. *Unsupervised Transfer Learning*

In unsupervised transfer learning, neither the source nor the target domain contains labeled data. The model identifies structural similarities between tasks, often for clustering, feature extraction, or anomaly detection. This is particularly useful for feature extraction in domains where labeled data is scarce but unlabeled data is available.

1) *Process:* Using unsupervised feature extraction, the model learns meaningful representations from the source data that can be used to extract or cluster relevant features in the target data. Techniques like autoencoders, GANs, or self-supervised learning are commonly employed to identify structural similarities and shared representations.
2) *Applications:* Found in clustering for social media analysis, where models can identify patterns or topics from one platform (e.g., Twitter) and apply these features to new, similar platforms (e.g., Reddit). In healthcare, unsupervised transfer learning can cluster patient data or extract features relevant to patient stratification or disease progression analysis.

D. *Transfer Learning in Practice*

Given the diversity of real-world applications, transfer learning has become an essential tool across fields such as image and speech recognition, predictive maintenance, robotics, and beyond:

1) *Image Recognition:* Pretrained convolutional neural networks (CNNs) like ResNet or VGG, trained on large datasets such as ImageNet, are fine-tuned on new image domains, such as medical or satellite images, where data labeling is costly. By leveraging general features like edges, textures, and object outlines from the source model, fine-tuning significantly enhances performance on the target dataset.
2) *Speech Recognition:* Transfer learning is applied in automatic speech recognition (ASR) where a model trained on one language can be adapted to another by reusing phonetic and acoustic patterns. This is valuable for developing ASR in languages with limited labeled data, reducing the need for extensive datasets by adapting from high-resource languages.
3) *Predictive Maintenance:* In industrial applications, data scarcity and differences in machinery setup make transfer learning ideal for predictive maintenance. A model trained on machinery in one factory can be adapted to another factory's machinery through domain adaptation techniques, improving efficiency by identifying failure patterns without extensive re-labeling.
4) *Robotics:* Transfer learning in robotics allows models to generalize across different environments, leveraging knowledge from simulation environments (source domain) to real-world tasks (target domain). This approach reduces the need for training directly on expensive or risky real-world tasks by using simulated training to learn basic task structures and then fine-tuning them in real-world scenarios.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

E. Technical Advancements Driving Transfer Learning

Advancements in deep learning frameworks and methodologies have enhanced transfer learning by optimizing both the efficiency and effectiveness of knowledge transfer:

1) *Domain Adaptation Techniques*: Techniques like MMD, adversarial training, and variational autoencoders (VAEs) aim to make learned features domain-agnostic, which is particularly beneficial in transductive and unsupervised transfer learning where domain shifts are prevalent.
2) *Self-supervised Learning:* Self-supervised learning techniques use large amounts of unlabeled data to learn representations through proxy tasks, which are then transferred to downstream tasks. Self-supervision has become especially impactful in unsupervised transfer learning for applications like language modeling and visual object recognition.
3) *Multi-Task Learning:* By training a single model to handle multiple related tasks (e.g., image captioning and object detection), multi-task learning promotes the learning of more generalized features, which can then be transferred to new, similar tasks more effectively.

F. Implications and Future Directions in Transfer Learning

Transfer learning is increasingly recognized for its capacity to bridge the gap between data-rich and data-scarce environments, addressing real-world challenges by reducing dependency on large labeled datasets. Its applications in low-resource settings, like healthcare and sustainability, demonstrate its potential for impactful contributions to society. Future research is exploring meta-transfer learning, where models are trained to quickly adapt to new tasks with minimal data, and few-shot learning, which enables effective learning with a few samples, further broadening the scope and feasibility of transfer learning across diverse applications.

IV. TRANSFER LEARNING CHARACTERISTICS

Transfer learning aims to repurpose a pre-trained model on a new but related task. Optimally selecting features and relevant variables is essential for improving model efficiency and performance. Feature selection algorithms, including filter, wrapper, and embedded methods, are commonly employed to refine data in both base and target datasets. Feature selection improves model performance by reducing overfitting, cutting down training time, and enhancing accuracy.

A. Datasets.

In transfer learning, a dataset is considered similar if the source and target domains align closely in structure or content. For example, the Places 205 Database ([Places 205] (http://places.csail.mit.edu/)) includes 2.5 million images across 205 scene categories, making it ideal as a base dataset for scene classification tasks. Transfer learning scenarios involving the Places 205 Database often use related target datasets, such as CS-280 Mini Places or Places365, which contain subsets that retain structural similarity. A contrasting example is ImageNet, which serves as a base dataset for various small target datasets, such as ImageNet8x8, ImageNet16x16, ImageNet32x32, and ImageNet64x64, each maintaining the same number of training images but differing in resolution. Smaller datasets like CIFAR-10 , CIFAR-100 , and STL-10 are also popular for transfer learning due to their manageable sizes (10–100 classes). Generally, when labeled data is scarce, feature engineering and transfer techniques become invaluable for model generalization.

B. Data Diversity

Data diversity plays a crucial role in determining if a dataset is suitable for transfer learning. For instance, datasets within a single industry (e.g., healthcare) might have high domain-specific alignment, but they may also have subtle links to other sectors, such as transportation (logistics of medical supplies) and banking (financial transactions in healthcare). Cross-domain data fusion, such as utilizing information from both healthcare and manufacturing sectors, could provide new perspectives for model training.

C. Dataset Size

In transfer learning, dataset size significantly impacts model training strategies. A "small" dataset typically has less than 25–30% of the classes compared to the base dataset. For instance, if using the Places 205 as a base, any dataset with fewer than 50 scene classes would be considered small. When the sample size per class exceeds 10,000 images, the dataset is considered large. Additionally, training datasets usually need to comprise at least 70% of the base dataset, ideally containing over 100 classes for effective knowledge transfer

D. Parameter Sharing

Parameter sharing allows the reuse of model parameters when processing target datasets of varying spatial dimensions. In transfer learning, convolutional and pooling layers can operate on inputs of different sizes without requiring extensive retraining, allowing the pretrained model to adapt to different spatial dimensions effectively.

E. Ensemble Methods

When there are distributional differences between training and test datasets, ensemble transfer learning techniques can enhance model performance. Ensemble methods integrate multiple models, thus improving classification accuracy by mitigating the effect of data insufficiencies, a common challenge in transfer learning scenarios with limited target data.

F. Data Epoch and Batch Processing

To handle large datasets, transfer learning models often divide data into batches, enabling sequential processing and weight updates. Batch sizes, iterations, and epochs control how many times the model adjusts based on data segments, allowing models to generalize well across large target datasets even when computational resources are limited.

G. Dataset Mapping

The effectiveness of transfer learning depends on the size and similarity of the target dataset to the base dataset. Transfer learning best practices include six common scenarios:

Expanding on each transfer learning scenario will help clarify the technical steps for implementing them based on dataset characteristics, such as similarity to the base dataset and size:

1) Scenario 1: Small, Similar Target Dataset to the Base Training Dataset

When working with a small and similar target dataset to the base dataset, the approach focuses on maximizing data efficiency without overfitting, given the limited amount of target data. A common strategy is **feature extraction**, where we leverage the pre-trained model's layers—especially the early and intermediate layers that contain generalized features learned from the base dataset. The process typically includes the following steps:

a) *Model Setup:* Retain most layers of the base model (e.g., ResNet or VGG), as the task similarity allows for high feature transferability, particularly in the shared domain features.

b) *Fine-tuning:* Fine-tune only the last few layers, which are more specific to the base dataset, to align them with the target dataset's subtle nuances. By freezing earlier layers, we reduce computational costs and avoid overfitting.

c) *Data Augmentation:* Due to the small dataset size, apply aggressive data augmentation techniques to artificially expand the dataset. This includes transformations like rotation, flipping, scaling, and cropping to help the model generalize better on the target data.

2) Scenario 2: Large, Similar Target Dataset to the Base Training Dataset

When the target dataset is large and similar to the base dataset, the focus is on optimizing for computational efficiency while taking advantage of the robust feature similarity. Here, the approach may involve deeper layers, as the larger dataset allows more extensive fine-tuning without overfitting.

a) *Extensive Fine-tuning*: Since the dataset is large, unfreezing a larger number of layers, including some middle and final layers, is beneficial. This allows the model to adjust to slight domain-specific nuances while still leveraging the foundational features from the base dataset.

b) *Batch Size and Learning Rate Adjustments*: Adjust batch sizes and learning rates according to dataset size, often using larger batch sizes to improve gradient estimates and faster convergence.

c) *Regularization Techniques*: Even with large datasets, regularization techniques such as dropout, batch normalization, and early stopping can help control overfitting and ensure robust generalization to the target data.

3) Scenario 3: Small, Different Target Dataset from the Base Training Dataset

A small target dataset that differs significantly from the base dataset poses a unique challenge. Here, the pre-trained model provides a starting point, but substantial fine-tuning or modification is required to bridge the domain gap.

a) *Feature Extraction with Custom Layers*: Use the pre-trained model as a fixed feature extractor by freezing the base layers. However, because the domains differ, add custom layers (e.g., fully connected layers or attention mechanisms) that learn specific features relevant to the target dataset.
b) *Transfer Learning with Domain Adaptation*: Domain adaptation techniques such as Maximum Mean Discrepancy (MMD) can help align features from the base and target datasets, reducing domain discrepancy without requiring large-scale data from the target domain.
c) *Aggressive Data Augmentation and Synthetic Data Generation*: Given the small dataset size, data augmentation is crucial, along with potential synthetic data generation using techniques like GANs or data mixing. This artificially expands the dataset and provides greater variability, aiding in domain-specific feature learning.

4) *Scenario 4: Large, Different Target Dataset from the Base Training Dataset*

A large and different target dataset allows flexibility for model adjustment and fine-tuning. This scenario benefits from the extensive capacity of the pre-trained model but requires more customization to accommodate the new domain.

a) *Extensive Layer Unfreezing and Fine-tuning:* With a large target dataset, more layers in the pre-trained model, including lower-level layers, can be unfrozen and fine-tuned, enabling the model to capture domain-specific patterns across all layers.
b) *Custom Architecture Modifications:* Since the domains differ, custom architectural adjustments, such as adding specialized convolutional or attention layers, can help the model focus on unique domain features.
c) *Domain-Specific Regularization Techniques:* Regularization methods like label smoothing and model assembling can improve generalization across a diverse target dataset, while transfer learning techniques such as progressive resizing—training with smaller images initially, then fine-tuning with full-sized images—can improve training efficiency.

5) *Scenario 5: Initializing with a pre-trained Network Instead of Random Initialization*

Initializing a model with pre-trained weights rather than random initialization is foundational to transfer learning, leveraging existing knowledge from a model trained on a large base dataset. This approach accelerates convergence and enhances accuracy, particularly when the target task shares underlying features with the base task.

a) *Weight Transfer*: Start with pre-trained weights on the entire network, transferring knowledge from a broad base (such as ImageNet) to the target task.
b) *Layer-specific Learning Rate Scheduling:* Since pre-trained weights provide a stable starting point, use layer-specific learning rates, where earlier layers have lower rates, and fine-tuned layers have higher rates.
c) *Benefits for Computational Efficiency:* Initializing with pre-trained weights shortens training time and enables the model to reach optimal performance faster than random initialization, especially on complex tasks with limited labeled data.

6) *Scenario 6: Freezing Weights of All but the Final Layer, Fine-tuning Only the Last Layer*

Freezing weights for all layers, except the last fully connected layer, is a straightforward transfer learning approach suited for scenarios where the target dataset is small or differs minimally in structure.

a) *Weight Freezing:* All layers, except the final layer, retain their pre-trained weights, preserving the general feature knowledge. This is effective for tasks that don't deviate significantly from the base task.
b) *Replacing the Final Layer*: Replace the final layer with a new fully connected layer initialized with random weights, which then adapts specifically to the target task by learning to map high-level features to the target classes.
c) *Fine-tuning and Optimization*: Only the final layer is fine-tuned, using a higher learning rate compared to what would be used in more extensive fine-tuning. This approach minimizes overfitting while still adapting to the target task.

In deep learning, pre-trained models are often initialized with general features in the first layer, such as Gabor filters and color blobs, which are applicable across various datasets and domains. These general features serve as a foundational layer for transfer learning, allowing target domain-specific features to be trained more efficiently. For example, a model pre-trained on a Places dataset can use its learned edge and texture features for similar datasets, expediting the training process while reducing resource requirements. By leveraging pre-trained models, the time and computational resources needed to extract general features are significantly reduced, streamlining model development

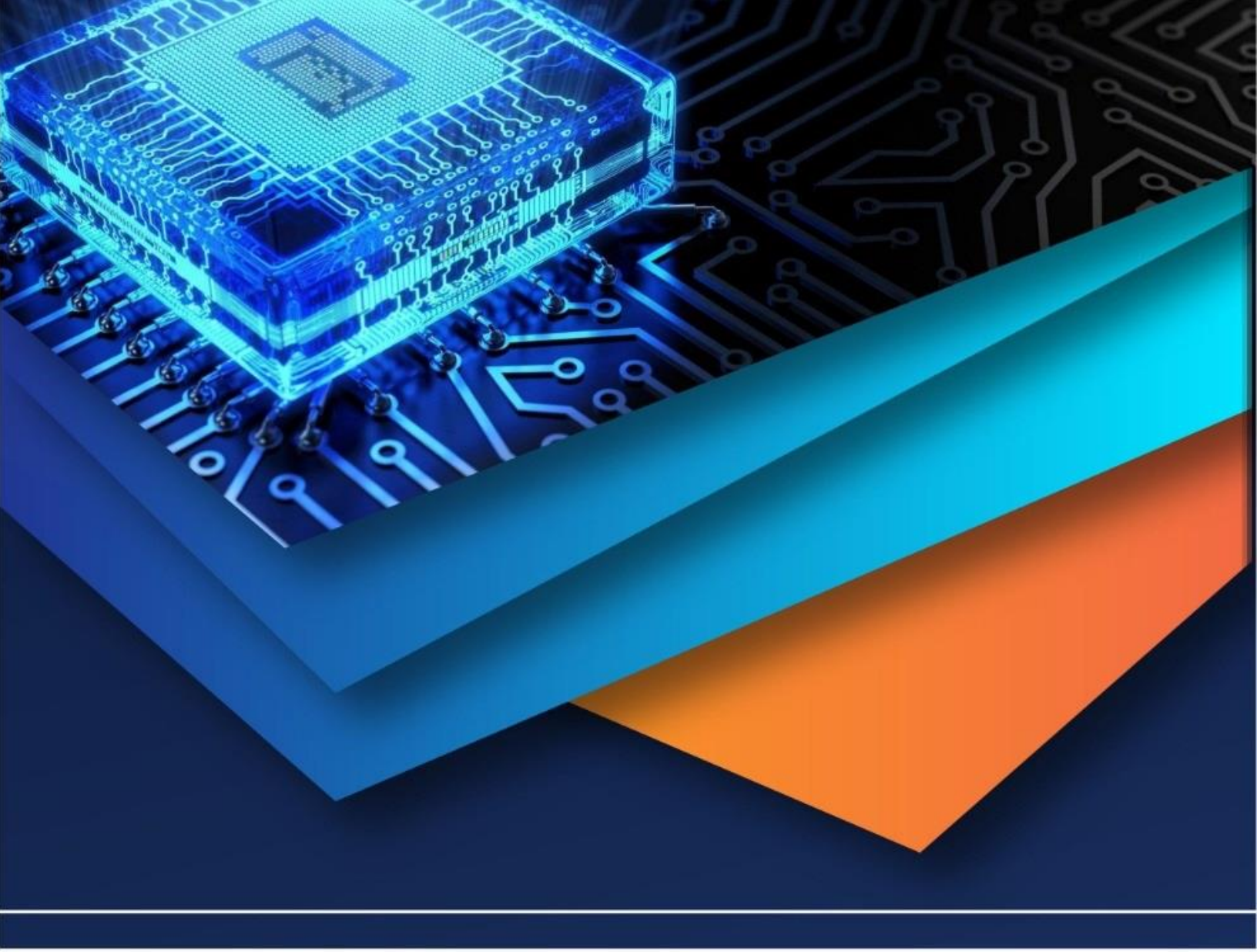
V. CONCLUSION

In this paper, we observed that building a machine-learning model from scratch is often resource-intensive, demanding significant time and computational power, especially in environments with limited hardware or software resources.

Transfer learning offers a valuable solution by significantly improving model performance, accuracy, and learning efficiency, as it leverages knowledge from pre-trained models rather than starting anew. By accelerating training time, transfer learning enables further exploration into optimizing learning rates and enhancing model robustness. It also provides measurable accuracy gains over baseline performance, with established models like ResNet50 [31], VGG16 [32], VGG19 [33], and InceptionV3 [34] demonstrating reliable improvements in diverse applications.

REFERENCES

[1] Rina Dechter (1986). Learning while searching in constraint-satisfaction problems. University of California, Computer Science Department, Cognitive Systems Laboratory.
[2] Ivakhnenko, Alexey (1971). "Polynomial theory of complex systems". IEEE Transactions on Systems, Man andCybernetics. 1(4): 364–378.
[3] Fukushima, K. (1980). "Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position". Biol. Cybern. 36 (4): 193–202.
[4] LeCun et al., "Backpropagation Applied to Handwritten Zip Code Recognition," Neural Computation, 1, pp. 541–551, 1989.
[5] Pratt, L. Y. (1993). "Discriminability-based transfer between neural networks" (PDF). NIPS Conference: Advances in Neural Information Processing Systems 5. Morgan Kaufmann Publishers. pp. 204–211
[6] Baxter, J., "Theoretical Models of Learning to Learn", pp. 71-95 Pratt & Thrun 1998
[7] Pratt, L. (1996). "Special Issue: Reuse of Neural Networks through Transfer". Connection Science. Retrieved 2017-08-10.
[8] Mehryar Mohri, Afshin Rostamizadeh, Ameet Talwalkar (2012) Foundations of Machine Learning, The MIT Press ISBN 9780262018258
[9] Jordan, Michael I.; Bishop, Christopher M. (2004). "Neural Networks". In Allen B. Tucker. Computer Science Handbook, Second Edition (Section VII: Intelligent Systems). Boca Raton, Florida: Chapman & Hall/CRC Press LLC. ISBN 1-58488- 360-X.
[10] Claude Sammut, and Geoffrey I. “Encyclopedia of Machine Learning (pp.159-162)”
[11] Yoshua Bengio, Ian J. Goodfellow, Aaron Courville (2015) “Deep Learning (pp.183-200)”
[12] Marcel van Gerven, Sander Bohte "Artificial Neural Networks as Models of Neural Information Processing | FrontiersResearch Topic"
[13] Indra Den Bakker “Python Deep Learning Cookbook, Packt Publishing (pp.173-189)” ISBN 978-1-78712-519-3
[14] Yoshua Bengio, Ian J. Goodfellow, Aaron Courville (2015) “Deep Learning (pp.382-384)”
[15] Giancarlo Zaccone, Md. Rezaul Karim, Ahmed Menshawy “Deep Learning with TensorFlow, (pp.98)” ISBN 978-1-78646-978-6
[16] Claude Sammut, and Geoffrey I. “Encyclopedia of Machine Learning (pp.99)”
[17] Goodfellow, Ian; Pouget-Abadie, Jean; Mirza, Mehdi; Xu, Bing; Warde-Farley, David; Ozair, Sherjil; Courville, Aaron; Bengio, Joshua (2014). "Generative Adversarial Networks".
[18] Kohonen, Teuvo; Honkela, Timo (2007). "Kohonen Network". Scholarpedia (http://www.scholarpedia.org/article/Kohonen_network)
[19] Emilio Soria Olivas, José David Martín Guerrero, Marcelino Martinez Sober, Jose Rafael Magdalena Benedito, Antonio José Serrano López, “Handbook of Research on Machine Learning Applications and Trends: Algorithms, Methods, and Techniques (pp.242-264)” Information Science Reference. ISBN 978-1-60566-767-6
[20] Emilio Soria Olivas, José David Martín Guerrero, Marcelino Martinez Sober, Jose Rafael Magdalena Benedito, Antonio José Serrano López, “Handbook of Research on Machine Learning Applications and Trends: Algorithms, Methods, and Techniques (pp.245-246)” Information Science Reference. ISBN 978-1-60566-767-6
[21] A. Arnold, R. Nallapati, and W. W. Cohen, “A comparative study of methods for transductive transfer learning,” in Proceedings of the 7thIEEE International Conference on Data Mining Workshops. Washington, DC, USA: IEEE Computer Society, 2007, pp. 77–82.
[22] Sinno Jialin Pan, Qiang Yang, “A Survey of Transfer Learning https://ieeexplore.ieee.org/document/5288526/”
[23] Lisa Torrey and Jude Shavlik “Transfer Learning”, University of Wisconsin, Madison, WI, USA.
[24] Place 205 Dataset - http://places.csail.mit.edu/user/download.php
[25] Miniplace - https://www.kaggle.com/c/cs280-mini-places/rules
[26] Places365 - http://places2.csail.mit.edu/download.html
[27] Image Net - http://image-net.org/index
[28] CIFER-10 Dataset - https://www.cs.toronto.edu/~kriz/cifar.html
[29] CIFER-100 Dataset - https://www.cs.toronto.edu/~kriz/cifar.html
[30] STL-10 Dataset - https://cs.stanford.edu/~acoates/stl10/
[31] ResNet50 - https://www.kaggle.com/dansbecker/transfer-learning/data
[32] VGG16 Model in Kaggle - https://www.kaggle.com/keras/vgg16
[33] VGG19 Model in Kaggle - https://www.kaggle.com/keras/vgg19
[34] InceptionV3 Model in Kaggle - https://www.kaggle.com/google-brain/inception-v3
[35] Souza, M.D., Prabhu, G.A., Kumara, V. et al. EarlyNet: a novel transfer learning approach with VGG11 and EfficientNet for early-stage breast cancer detection. Int J Syst Assur Eng Manag (2024). https://doi.org/10.1007/s13198-024-02408-6
[36] Melwin D'souza, Ananth Prabhu Gurpur, Varuna Kumara, “SANAS-Net: spatial attention neural architecture search for breast cancer detection”, IAES International Journal of Artificial Intelligence (IJ-AI), Vol. 13, No. 3, September 2024, pp. 3339-3349, ISSN: 2252-8938, DOI: http://doi.org/10.11591/ijai.v13.i3.pp3339-3349
[37] Melwin D Souza, Ananth Prabhu G and Varuna Kumara, A Comprehensive Review on Advances in Deep Learning and Machine Learning for Early Breast Cancer Detection, International Journal of Advanced Research in Engineering and Technology (IJARET), 10 (5), 2019, pp 350-359.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)