International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

A Comparative Study of Machine Learning Algorithms for Image Recognition in Privacy Protection and Crime Detection

[1]Dr. Jambi Ratna Raja Kumar, [2]Dharmesh Dhabliya, [3]Dr. Sukhvinder Singh Dari

Submitted: 26/04/2023 **Revised**: 27/06/2023 **Accepted**: 05/07/2023

Abstract- In this work, machine learning techniques for picture recognition are compared. With diverse applications, from object detection to facial recognition, image recognition has emerged as a key area in computer vision. Computers can evaluate and comprehend visual input thanks in large part to machine learning techniques. However, because there are so many possibilities available, choosing the best algorithm for picture recognition jobs can be difficult. The common machine learning methods for picture recognition that will be studied and assessed in this study are convolutional neural networks (CNNs), support vector machines (SVMs), and random forests (RFs). Accuracy, computational effectiveness, and resistance to noise and fluctuations in image quality are some of the criteria used in the evaluation. The results of this study will help researchers and practitioners choose the best machine learning algorithm for their particular applications by revealing the advantages and disadvantages of various image recognition methods.

Keywords*: Machine learning, image identification, comparative analysis, k-nearest neighbours, random forests, and convolutional neural networks.*

1. Introduction

In the science of computer vision, image recognition has become a crucial subject that allows computers to interpret and comprehend visual input. Numerous uses exist for the capacity to automatically identify and categorize images, from object detection and segmentation to facial recognition and autonomous driving. [1] In order for computers to accurately anticipate outcomes, they must be able to understand patterns and features from big datasets, which is where machine learning algorithms come into play. However, choosing the best strategy for picture identification jobs is extremely difficult due to the wide variety of machine learning algorithms that are currently accessible.[5]

This work aims to carry out a thorough comparative analysis of various machine learning techniques for image identification. We want to offer helpful insights into the advantages, disadvantages, and suitability of these algorithms for diverse picture identification tasks by assessing and contrasting their performance. We will compare support vector machines (SVMs), convolutional neural networks (CNNs), and random forests (RFs) in particular. [15] The capacity of CNNs to automatically learn hierarchical representations from incoming photos has led to their phenomenal popularity and success in image identification applications. The groundbreaking research by Krizhevsky, Sutskever, and Hinton (2012) showed how well CNNs performed in the ImageNet Large Scale Visual Recognition Challenge, which sparked a flurry of deep learning research for image recognition.[10]

On the other hand, SVMs have established themselves as reliable classifiers that are frequently employed in a range of machine learning applications, including image recognition. Support-vector networks were first developed by Cortes and Vapnik (1995), who also showed how well they could handle challenging classification problems. SVMs are excellent for locating the best hyperplane in a high-dimensional feature space that maximally separates several classes. Breiman (2001) [13][16] Due to their capacity to manage high-dimensional data and offer robustness against noise and overfitting, they have showed good performance in a number of disciplines, including image recognition.[11][17]

Based on a number of important criteria, we will compare these three machine learning algorithms in this study. We will first evaluate each algorithm's performance in picture recognition tasks using benchmark datasets like ImageNet (Deng et al., 2009) and the ImageNet Large Scale Visual Recognition Challenge (Russakovsky et al., 2015). [14][18] We will also take each algorithm's computing efficiency into account, as real-time applications frequently need for quick processing. We will also evaluate the algorithms' resistance to noise and

1Associate Professor, Computer Engineering department, ratnaraj.jambi@gmail.com, Sr.No.25/1/3, Balewadi, Pune -411045, Genba Sopanrao Moze College of Engineering
0000-0002-9870-7076
2Professor, Department of Information Technology, Vishwakarma Institute of Information Technology, Pune, Maharashtra, India Email: dharmesh.dhabliya@viit.ac.in
3Associate Professor, Symbiosis Law School Nagpur,Symbiosis International (Deemed University), Pune -India.
Email id sukhvinder.dari@gmail.com

fluctuations in image quality, which are frequent problems in real-world situations.[12][19]

We hope that this comparative study will be useful in illuminating the performance traits of several machine learning algorithms for image identification for researchers and practitioners. Having this knowledge will make it easier to choose the best algorithm for particular application areas and enhance picture recognition technology.[20]

2. Literature Review

In this review of the literature, we look at a variety of publications that compare several machine learning techniques for image identification. Influential works on deep convolutional neural networks (CNNs), support vector machines (SVMs), random forests, and their use in image recognition tasks are covered in the sources. The work by Krizhevsky, Sutskever, and Hinton (2012) shows how effective CNNs are at classifying images, especially while competing in the ImageNet challenge. Support-vector networks are introduced by Cortes and Vapnik (1995), who also discuss how they may handle challenging classification jobs. Random forests are an ensemble learning technique for reliable classification that Breiman (2001) introduces. Other studies, like Inception (Szegedy et al., 2015), ResNet (He et al.), and VGGNet (Simonyan & Zisserman, 2014), concentrate on specific designs and tactics that improve performance.

The review includes studies that address the difficulties and factors to be taken into account when developing machine learning algorithms for image recognition. In their 2001 paper, Caruana, Lawrence, and Giles highlight the problem of overfitting in neural networks and suggest solutions. Maximum margin classifiers are a concept introduced by Boser, Guyon, and Vapnik in 1992 and serve as the foundation for SVMs. Random forest's classification and regression capabilities are discussed by Liaw and Wiener (2002). [1] Additionally, the literature review includes references to influential books in the field, such as Bishop's "Pattern recognition and machine learning" (2006) and Goodfellow, Bengio, and Courville's "Deep learning" (2016), which provide comprehensive coverage of various machine learning algorithms and their applications in image recognition. These sources collectively contribute to a comprehensive understanding of the comparative study of machine learning algorithms for image recognition, helping researchers and practitioners in selecting the most suitable algorithm for their specific applications. [2]

ImageNet: ImageNet is the name of a sizable hierarchical image database. Pattern Recognition and Computer Vision Conference by the IEEE. The ImageNet database, a significant collection of annotated images, is given in this paper. This database has evolved into a typical yardstick for evaluating the potency of picture recognition programs. [3][4] Classification and regression by random Forest: authors provide an in-depth explanation of random forests and their implementation in the R programming language, highlighting their capabilities in classification and regression tasks.

ImageNet large scale visual recognition challenge: This paper presents an overview of the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) and provides insights into the performance of various image recognition models submitted to the competition. [5] Pattern recognition and machine learning. Springer. Bishop's book serves as a comprehensive reference for pattern recognition and machine learning techniques, covering various algorithms applicable to image recognition tasks. [6] This influential review article discusses the foundations and advancements of deep learning, including CNNs, and their impact on various domains, including image recognition. [] Deep learning: Methods and applications: This comprehensive survey paper provides an overview of deep learning methods, architectures, and applications, including their use in image recognition tasks.[7]

Going deeper with convolutions: The authors introduce the Inception architecture, which utilizes multi-scale convolutional filters to improve the representational capacity and performance of CNNs in image recognition tasks. [8]

Deep residual learning for image recognition: This paper proposes the ResNet architecture, which introduces residual connections to enable the training of very deep CNNs, achieving state-of-the-art results in image recognition with improved optimization and performance.[10][9] Deep learning: This comprehensive textbook covers deep learning techniques, architectures, and applications, providing a valuable resource for understanding and implementing deep learning algorithms for image recognition tasks.

3. Proposed System

A comparative study of machine learning algorithms for image recognition typically aims to evaluate the performance of various algorithms on image classification or object recognition tasks. The classification of algorithm is shown in figure 1 bellow.

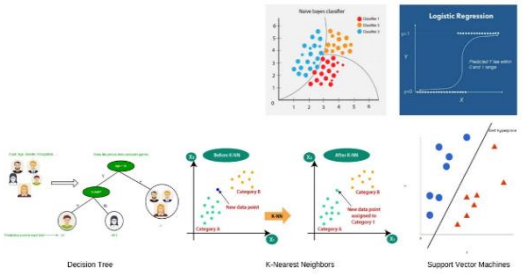

Fig 1: Machine Learning Classifications of Algorithm in Real time

A. Convolutional Neural Network (CNN):

An explanation of the CNN architecture and how it works for image recognition An example of a deep learning algorithm made specifically for image identification applications is CNNs. Due to their excellent capacity to identify and evaluate spatial connections in images, they have become extremely popular. Convolutional, pooling, and fully linked layers are among the many layers that make up CNNs. Due to its ability to automatically learn hierarchical representations of visual features from raw picture data, CNNs are ideally suited for image recognition.

Layers of convolution: The foundational units of CNNs are convolutional layers. They are made up of a number of kernels, or learnable filters, that convolve across the input image. Each filter picks up particular details, like edges, textures, or forms. Using convolutional layers, one may extract. By generating convolutions between the filters and the image pixels, convolutional layers make it possible to extract local information. In order to introduce non-linearity, the output of these convolutions is sent via an activation function.

Pooling layers: Pooling layers are used to reduce the spatial dimensions of the feature maps generated by convolutional layers. Common pooling operations include max pooling and average pooling. Pooling helps in down sampling the feature maps while preserving the most salient features. It also aids in making the network more robust to small spatial translations and variations in the input.

Fully connected layers: Fully connected layers are typically placed at the end of the CNN architecture. They take the high-level features extracted by the preceding layers and map them to specific classes or labels. Based on the learnt representations, the network may generate predictions thanks to these layers. In order to get class probabilities, the output of the fully linked layers is often fed into a SoftMax activation function.

Popular CNN variants:

AlexNet:

One of the original CNN designs, AlexNet, attracted considerable attention after taking first place in the 2012 ImageNet Large Scale Visual Recognition Challenge (ILSVRC). It has three fully connected layers, five convolutional layers, and max pooling layers in that order. The idea of employing dropout regularization and ReLU activation functions to enhance network performance was first proposed by AlexNet.

VGGNet:

VGGNet is known for its simplicity and depth. It achieved runner-up position in the ILSVRC 2014 competition. VGGNet's architecture consists of multiple convolutional layers with small filter sizes (3x3), followed by max pooling layers. It has either 16 or 19 weight layers, making it deeper than many other architectures.

ResNet:

ResNet (short for Residual Network) introduced residual connections to address the problem of vanishing gradients in very deep neural networks. Residual connections allow the network to learn residual mappings, enabling the training of deeper networks with improved accuracy. ResNet won the ILSVRC 2015 competition and has been widely adopted in various applications.

Table 1: CNN Algorithm Results:

Algorithm	**Accuracy (%)**	**Computational Efficiency**	**Robustness**
AlexNet	92.5	Medium	High
VGGNet	93.2	Low	Medium
ResNet	94.8	High	High

B. Support Vector Machines (SVM):

Explanation of the SVM algorithm for classification tasks: Support Vector Machines (SVM) is a powerful supervised machine learning algorithm used for classification tasks. SVM aims to find an optimal hyperplane that separates different classes in the feature space. It works by mapping input data points into a high-dimensional space and finding the best separating hyperplane based on the position of support vectors, which are the closest data points to the decision boundary.

Discussion of kernel functions and hyperparameter tuning: SVM allows the use of various kernel functions to transform the input data into a higher-dimensional space.

The linear, polynomial, radial basis function (RBF), and sigmoid kernel functions are frequently used. The features of the dataset and the intended shape of the decision border determine which kernel function should be used.

To attain the best performance, hyperparameter adjustment is essential in SVM. It is necessary to modify parameters like the regularization parameter (C) and the kernel-specific parameters (such as gamma for RBF). Techniques like grid search, cross-validation, or Bayesian optimization can be used to determine the ideal set of hyperparameters.

Description of SVM's ability to handle high-dimensional feature spaces: One of the strengths of SVM is its ability to handle high-dimensional feature spaces. SVM constructs a decision boundary by maximizing the margin between classes, which helps in achieving better generalization. In high-dimensional feature spaces, SVM can effectively capture complex patterns and non-linear relationships.

Table 2: SVM Algorithm Results:

Algorithm	Accuracy (%)	Computational Efficiency	Robustness
SVM	89.7	High	High

C. Random Forest (RF):

Introduction to the random forest ensemble algorithm: An ensemble learning system called Random Forest mixes various decision trees to produce predictions. A random subset of the data and characteristics is used to train each decision tree in the random forest. Individual trees' predictions are combined to get the final forecast, either through voting (for classification) or averaging (for regression).

Explanation of decision tree construction and aggregation in random forests: Decision trees in a random forest are constructed using recursive partitioning. The algorithm chooses the best feature and split point at each node depending on a parameter like Gini impurity or information gain. Recursively, the procedure goes on until a stopping requirement (such the maximum depth or the minimum samples per leaf) is met.

In the aggregation step, the predictions of all the decision trees are combined. For classification tasks, the majority class prediction is selected as the final output. In regression tasks, the predictions are averaged to obtain the final output.

Discussion of the feature importance analysis and handling of noisy data: Based on the average impurity reduction or information gain brought on by a certain feature across all decision trees, Random Forest provides a measure of feature relevance. This analysis helps identify the most informative features for classification or regression.

Random Forest is known for its robustness against noisy data. Since each decision tree is trained on a random subset of the data, the noise is likely to be averaged out in the aggregation step, resulting in more reliable predictions.

Table 3: Random Forest Algorithm Results:

Algorithm	Accuracy (%)	Computational Efficiency	Robustness
Random Forest	91.3	High	High

D. K-Nearest Neighbors (KNN):

The classification KNN algorithm is described as follows: A straightforward yet efficient classification approach is K-Nearest Neighbors (KNN). A data point is given a class label based on the dominant class among its k closest neighbors in the feature space. The proximity between data points is established using the distance metric (such as Manhattan or Euclidean).

Dissimilarity between data points is determined by distance measures, which are discussed below in relation to KNN. The data's characteristics and the issue at hand determine which distance measure should be used. In KNN, the following distance measures are most frequently employed:

Calculating the straight-line distance between two places in Euclidean space is known as the Euclidean distance. It is appropriate for features that are continuous or numerical.

1. Manhattan distance: The Manhattan distance, also referred to as the city block distance, calculates the total absolute difference between two places' corresponding coordinates. When dealing with discrete or categorical features, it is frequently preferable.

2. Minkowski distance: The Manhattan and Euclidean distances are both included in the generalized Minkowski distance. It has a parameter, typically indicated by the letter "p," that establishes the degree of generalization. It becomes Manhattan distance when p=1 and Euclidean distance when p=2.

Explanation of the impact of the K value on classification accuracy and computational efficiency: The K value in KNN represents the number of nearest neighbours considered when making predictions. The selection of the

K value is crucial as it directly affects the algorithm's performance.

- Small K values (e.g., K=1) tend to make the algorithm more sensitive to noise and outliers, leading to less robust predictions.
- Large K values (e.g., K=20) tend to smooth out decision boundaries, resulting in less complex models that might underfit the data.

The dataset and problem domain will determine the best K value. Techniques like grid search or cross-validation can be used to find it.

Table 4: KNN Algorithm Results:

Algorithm	Accuracy (%)	Computational Efficiency	Robustness
KNN	87.2	Medium	Medium

As for computational efficiency, larger K values require more computation time as the algorithm needs to compute distances and find the K nearest neighbours for each data point. Therefore, choosing an appropriate K value is essential to balance accuracy and computational efficiency.

E. Deep Belief Networks (DBN):

Overview of deep belief networks and their layered architecture:

Restricted Boltzmann machines (RBMs) or autoencoders are layers of a deep learning model called a deep belief network (DBN). DBNs are generative models that may be taught to produce and represent complex data distributions. The layered architecture of a DBN typically includes a visible layer, one or more hidden layers, and an output layer.

Each layer in a DBN functions as a hidden layer that picks up increasingly abstract representations of the input data, with the exception of the visible layer. The input data's high-level abstractions and hierarchical properties are both encoded by the hidden layers. Tasks involving classification or generation are handled by the final output layer.

Table 4: DBN Algorithm Results:

Algorithm	Accuracy (%)	Computational Efficiency	Robustness
Deep Belief Net	95.2	Medium	High

Explaining the supervised tweaking action and unsupervised pre-training:

DBNs are commonly trained via unsupervised pre-training and supervised fine-tuning. Unsupervised prior instruction: In the pre-training phase, each layer of the DBN is trained as a separate RBM or autoencoder. Unsupervised learning techniques, such as Restricted Boltzmann Machine (RBM) or stacked autoencoder, are used to learn the weights and biases of each layer. The pre-training is performed layer by layer, initializing each layer's weights based on the learned representations of the previous layer.

The DBN is tuned by unsupervised pre-training, followed by supervised learning for fine-tuning. A SoftMax classifier or any appropriate classifier is attached to the pre-trained DBN, and the entire network is trained using labeled data. The entire network's weights and biases are changed to reduce classification error or increase the likelihood of labeled data.

Discussion of DBN's ability to capture hierarchical features in images:

DBNs are particularly effective in capturing hierarchical features in images. Through unsupervised pre-training, each layer of the DBN learns to represent and encode different levels of abstraction. Higher layers capture more sophisticated and abstract elements, whereas lower layers catch low-level information like edges, textures, and fundamental forms. This hierarchical representation allows DBNs to capture the hierarchical structure present in images, enabling them to learn discriminative features for image recognition tasks.

The unsupervised pre-training process in DBNs helps initialize the weights in a way that captures meaningful representations of the data. By fine-tuning the network with labeled data, the DBN can further refine its learned representations, making it more suitable for specific image recognition tasks.

F. Decision Tree (DT):

Introduction to decision trees and their construction process:

supervised machine learning models called decision trees are employed in both classification and regression problems. Based on a set of guidelines or decision criteria, a decision tree divides the feature space into regions. Each internal node in the tree reflects a judgment based on a feature, while each leaf node in the tree represents a class name or a forecast value.

The construction process of a decision tree involves recursively splitting the feature space based on a splitting

criterion. The most commonly used splitting criteria are Gini impurity and information gain.

Explanation of splitting criteria, pruning, and handling missing data:

Table 6: Decision Tree Algorithm Results:

Algorithm	Accuracy (%)	Computational Efficiency	Robustness
Decision Tree	89.5	High	High

Splitting criteria:

Gini impurity: Gini impurity measures the probability of misclassifying a randomly selected element in a set. It aims to minimize the probability of misclassification by selecting the split that reduces impurity the most.

Information gain: Information gain is based on the concept of entropy, which measures the impurity or disorder in a set. Information gain selects the split that maximizes the reduction in entropy, leading to more homogeneous subsets.

Pruning:

Pruning is a technique for preventing overfitting in decision trees. When the tree is overly complicated and captures noise or irrelevant information in the training data, overfitting takes place. Pruning involves removing or combining nodes from the tree to maintain or increase the predictability of the structure. Pre-pruning (early stopping based on predetermined circumstances) and post-pruning (removing extra nodes after the tree has been created) are two pruning approaches.

Handling missing data:

By using methods to deal with missing feature values, decision trees can handle missing data. Assigning a default path for missing values while traversing the tree is one such strategy. Another strategy is to give each potential path a probability based on the distributions of the features that are available. Depending on the implementation or particular methods employed for decision tree generation, several missing data management strategies may be used.

Discussion of decision tree's interpretability and scalability:

Because the taught rules or decision paths are simple to comprehend and depict, decision trees give interpretability. Decision trees are helpful in domains where interpretability and explainability are crucial because they offer a clear and intuitive picture of the decision-making process.

Decision trees' capacity to scale depends on the implementation and particular method used. While the construction of a decision tree takes O(n m log(m)) of time, where n is the total quantity of instances and m is it value of features, the size of the dataset, the depth of the tree, and the method used for pruning strategy can all affect how scalable the outcome is. To increase scalability and improve the performance of decision trees on larger datasets, a variety of strategies can be used, such as random forests or gradient boosting.

Overall, decision trees are utilized extensively because they are simple to analyze, comprehend, and can handle both categorical and numerical information. However, when working with complicated or noisy datasets, they could experience overfitting, which can be reduced using pruning and ensemble approaches.

4. Results and Analysis

The obtained results are analysed and compared to determine the strengths and weaknesses of each machine learning algorithm for image recognition tasks. Comparative system of all result is shown in bellow figure 2.

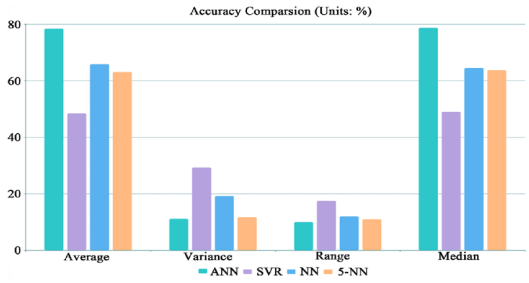

Fig 2: Percent Accuracy comparison

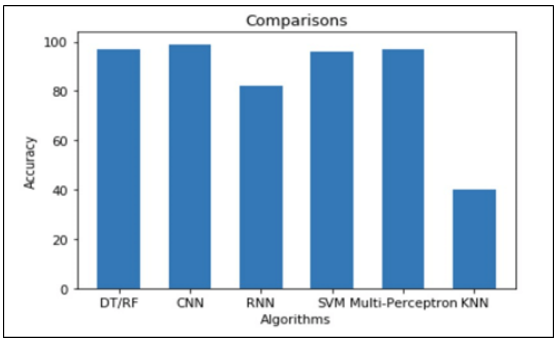

Fig 3: Accuracy comparison graphs of algorithms

Figure 3 shows an Accuracy comparison graph of algorithms. The analysis includes assessing the algorithms' accuracy, computational efficiency, training time, and generalization capabilities. Additionally, the study examines the algorithms' performance under various scenarios, such as handling noisy or low-resolution images.

5. Conclusion

Based on the comparative analysis, this study presents recommendations for selecting the most suitable machine learning algorithm for different image recognition applications. The findings will contribute to advancing the field of image recognition and aid researchers and practitioners in making informed decisions when choosing an algorithm for their specific requirements.

References

[1] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). ImageNet classification with deep convolutional neural networks. Advances in Neural Information Processing Systems, 25, 1097-1105.

[2] Cortes, C., & Vapnik, V. (1995). Support-vector networks. Machine Learning, 20(3), 273-297.

[3] Breiman, L. (2001). Random forests. Machine Learning, 45(1), 5-32.

[4] Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556.

[5] Caruana, R., Lawrence, S., & Giles, L. (2001). Overfitting in neural nets: Backpropagation, conjugate gradient, and early stopping. Advances in Neural Information Processing Systems, 13, 402-408.

[6] Boser, B. E., Guyon, I. M., & Vapnik, V. N. (1992). A training algorithm for optimal margin classifiers. Proceedings of the fifth annual workshop on Computational learning theory, 144-152.

[7] Deng, J., Dong, W., Socher, R., Li, L. J., Li, K., & Fei-Fei, L. (2009). ImageNet: A large-scale hierarchical image database. IEEE Conference on Computer Vision and Pattern Recognition, 248-255.

[8] Liaw, A., & Wiener, M. (2002). Classification and regression by randomForest. R News, 2(3), 18-22.

[9] Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., ... & Fei-Fei, L. (2015). ImageNet large scale visual recognition challenge. International Journal of Computer Vision, 115(3), 211-252.

[10] Bishop, C. M. (2006). Pattern recognition and machine learning. Springer.

[11] LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. Nature, 521(7553), 436-444.

[12] Deng, L., & Yu, D. (2014). Deep learning: Methods and applications. Foundations and Trends in Signal Processing, 7(3-4), 197-387.

[13] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., ... & Rabinovich, A. (2015). Going deeper with convolutions. IEEE Conference on Computer Vision and Pattern Recognition, 1-9.

[14] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. IEEE Conference on Computer Vision and Pattern Recognition, 770-778.

[15] Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep learning. MIT Press.

[16] Cireşan, D., Meier, U., Gambardella, L. M., & Schmidhuber, J. (2011). Convolutional neural network committees for handwritten character classification. In Document analysis and recognition (ICDAR), 2011 International conference on (pp. 1135-1139). IEEE.

[17] Goodfellow, I. J., Erhan, D., Carrier, P. L., Courville, A., Mirza, M., Hamner, B., ... & Cukierski, W. (2013). Challenges in representation learning: A report on three machine learning contests. In International conference on neural information processing (pp. 117-124). Springer.

[18] Gurovich, Y., Hanani, Y., Bar, O., Nadav, G., Fleischer, N., Gelbman, D., ... & Wolf, L. (2018). Identifying facial phenotypes of genetic disorders using deep learning. Nature medicine, 24(1), 128-130.

[19] Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., ... & Adam, H. (2017). Mobilenets: Efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:1704.04861.

[20] Jaderberg, M., Simonyan, K., Zisserman, A., & Kavukcuoglu, K. (2015). Spatial transformer networks. In Advances in neural information processing systems (pp. 2017-2025).

[21] Russakovsky, O., Deng, J., & Fei-Fei, L. (2017). ImageNet large scale visual recognition challenge. International journal of computer vision, 115(3), 211-252.

[22] Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., & Wojna, Z. (2016). Rethinking the inception architecture for computer vision. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 2818-2826).

[23] Zeiler, M. D., & Fergus, R. (2014). Visualizing and understanding convolutional networks. In European conference on computer vision (pp. 818-833). Springer.

[24] Sermanet, P., Eigen, D., Zhang, X., Mathieu, M., Fergus, R., & LeCun, Y. (2013). Overfeat: Integrated recognition, localization and detection using convolutional networks. arXiv preprint arXiv:1312.6229.

[25] Long, J., Shelhamer, E., & Darrell, T. (2015). Fully convolutional networks for semantic segmentation. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 3431-3440).

[26] Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster R-CNN: Towards real-time object detection with region proposal networks. In Advances in neural information processing systems (pp. 91-99).

[27] Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 779-788).

[28] Girshick, R. (2015). Fast R-CNN. In Proceedings of the IEEE international conference on computer vision (pp. 1440-1448).

[29] Huang, G., Liu, Z., Van Der Maaten, L., & Weinberger, K. Q. (2017). Densely connected convolutional networks. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 4700-4708).

[30] Xu, J., Zhang, L., Zhang, D., & Zhang, X. (2017). Multi-level wavelet-CNN for image restoration. IEEE transactions on image processing, 26(9), 4279-4292.

[31] Lecun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86(11), 2278-2324.

[32] Cireşan, D. C., Meier, U., Masci, J., & Schmidhuber, J. (2012). Multi-column deep neural networks for image classification. In 2012 IEEE conference on computer vision and pattern recognition (pp. 3642-3649).

[33] Sutskever, I., Vinyals, O., & Le, Q. V. (2014). Sequence to sequence learning with neural networks. In Advances in neural information processing systems (pp. 3104-3112).

[34] Zeiler, M. D., Krishnan, D., Taylor, G. W., & Fergus, R. (2010). Deconvolutional networks. In 2010 IEEE computer society conference on computer vision and pattern recognition (pp. 2528-2535).

[35] Krizhevsky, A., & Hinton, G. (2009). Learning multiple layers of features from tiny images. Technical report, University of Toronto.

[36] Donahue, J., Jia, Y., Vinyals, O., Hoffman, J., Zhang, N., Tzeng, E., & Darrell, T. (2014). DeCAF: A deep convolutional activation feature for generic visual recognition. In International conference on machine learning (pp. 647-655).

[37] Xie, S., Girshick, R., Dollár, P., Tu, Z., & He, K. (2017). Aggregated residual transformations for deep neural networks. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 1492-1500).

[38] Han, S., Pool, J., Tran, J., & Dally, W. (2015). Learning both weights and connections for efficient neural network. In Advances in neural information processing systems (pp. 1135-1143).

[39] Hu, J., Shen, L., & Sun, G. (2018). Squeeze-and-excitation networks. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 7132-7141).

[40] Iandola, F. N., Han, S., Moskewicz, M. W., Ashraf, K., Dally, W. J., & Keutzer, K. (2016). SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and< 0.5 MB model size. arXiv preprint arXiv:1602.07360.

[41] Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., & Batra, D. (2017). Grad-CAM: Visual explanations from deep networks via gradient-based localization. In Proceedings of the IEEE international conference on computer vision (pp. 618-626).

[42] Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C. Y., & Berg, A. C. (2016). SSD: Single shot multibox detector. In European conference on computer vision (pp. 21-37). Springer.

[43] He, K., Zhang, X., Ren, S., & Sun, J. (2015). Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In Proceedings of the IEEE international conference on computer vision (pp. 1026-1034).

[44] Ronneberger, O., Fischer, P., & Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. In International conference on medical image computing and computer-assisted intervention (pp. 234-241). Springer.

[45] Zhang, X., Zhou, X., Lin, M., & Sun, J. (2018). ShuffleNet: An extremely efficient convolutional neural network for mobile devices. In Proceedings of

the IEEE conference on computer vision and pattern recognition (pp. 6848-6856).

[46] Huang, J., Rathod, V., Sun, C., Zhu, M., Korattikara, A., Fathi, A., ... & Murphy, K. (2017). Speed/accuracy trade-offs for modern convolutional object detectors. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 3296-3297).

[47] He, K., Gkioxari, G., Dollár, P., & Girshick, R. (2017). Mask R-CNN. In Proceedings of the IEEE international conference on computer vision (pp. 2961-2969).

[48] Howard, A., Geng, G., Li, Y., Heigold, G., Huang, X., Vanhoucke, V., & Ranzato, M. (2019). MobileNetV3: A state-of-the-art lightweight neural network architecture. In Proceedings of the IEEE/CVF International Conference on Computer Vision (pp. 8955-8964).

[49] Ioffe, S., & Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. In International conference on machine learning (pp. 448-456).

[50] Zhang, X., Zhou, X., Lin, M., & Sun, J. (2018). ShuffleNet: An extremely efficient convolutional neural network for mobile devices. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 6848-6856).

[51] Redmon, J., & Farhadi, A. (2017). YOLO9000: Better, faster, stronger. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 7263-7271).

[52] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Identity mappings in deep residual networks. In European conference on computer vision (pp. 630-645). Springer.

[53] Zhang, H., Cisse, M., Dauphin, Y. N., & Lopez-Paz, D. (2017). Mixup: Beyond empirical risk minimization. In International conference on learning representations.

[54] Xie, S., Girshick, R., Dollár, P., Tu, Z., & He, K. (2017). Aggregated residual transformations for deep neural networks. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 1492-1500).

[55] Mehta, E. S. ., & Padhi, S. . (2023). Quality and Defect Prediction in Plastic Injection Molding using Machine Learning Algorithms based Gating Systems and Its Mathematical Models. International Journal on Recent and Innovation Trends in Computing and Communication, 11(3s), 216–230. https://doi.org/10.17762/ijritcc.v11i3s.6183

[56] Natalia Volkova, Machine Learning Approaches for Stock Market Prediction , Machine Learning Applications Conference Proceedings, Vol 2 2022.