Volume (1) No (10), 2025
Journal of Global Research in Multidisciplinary Studies (JGRMS)
Review Paper/Research Paper
Available online at https://saanvipublications.com/journals/index.php/jgrms/index

Automated Image Preprocessing Pipelines for Large-Scale Datasets Based on Deep Learning Models

Dr. Jvalantkumar Kanaiyalal Patel
Assistant Professor
Shri Manilal Kadakia College of Commerce, Management,Science and Computer Studies
Ankleshwar
jvalant007@gmail.com

***Abstract*—Improving the effectiveness of deep learning (DL) for image classification on large datasets requires not only data preprocessing but also the right network designs. In this research, a new image classification system is introduced, which applies a modified Inception model to a subset of the ImageNet dataset containing 10 classes. Preparation of the data includes cleaning, augmentation, normalization, and feature extraction. These activities not only preserve the quality of data but also provide efficient input to the model. The modified Inception architectural changes enhance the original Inception network idea, thus increasing the power of the network for feature extraction and data classification. To evaluate the results, besides the modified Inception model, several well-known DL architectures like DenseNet, ResNet50, and VGG16 have been employed and compared using common performance metrics. The modified Inception model achieves better performance in the classification tasks than the baseline models. The accuracy is at 98.7%, the precision at 98.5%, the recall at 99%, and the F1-score at 98.6%. The results reported here have highlighted the role of architectural modifications and the impact of the thorough preprocessing pipeline in boosting the performance of image classification tasks on large datasets. Such a system offers a viable means to address challenging visual recognition problems.**

Keywords—Deep Learning, Automated Optimization, Automated Machine Learning, Parameter Optimization, Sequential-Model-Based Optimization, Image Classification.

I. INTRODUCTION

In various fields like medical imaging, remote sensing, surveillance, and industrial automation, a large number of images are available. Sorting and viewing these images quickly and efficiently becomes very important in such cases[1][2]. Images are very significant for computer vision systems nowadays. The system's capability to recognize and understand these images is dependent on their quality. However, real-world image data are usually accompanied by noise, blur, fluctuating illumination, and different quality levels. These issues can make it difficult for the models to learn. Hence, it is very important to make sure that the images are clear and similar when you train the model. This step is very important for the development of image-based intelligence systems that are able to work properly. Nevertheless, these databases usually have raw, unstructured, and mixed image data in them[3]. This facilitates having a large number of images for training and ensures that models work efficiently.

Automated preprocessing pipelines have the ability to eliminate noise, normalize data, scale numbers, and adjust colors without the intervention of humans[4][5][6]. Adaptive algorithms modify how these systems examine images before sending them to the computer, depending on what they are. It thus makes things more efficient and stable. They can, by the use of these methods, very quickly prepare a huge number of images for the following stages[7][8]. Such pipelines often perform operations like noise reduction, normalization, scaling, augmentation, and feature enhancement. These techniques turn huge mixed image sets into manageable ones which would be unfeasible to handle by hand[9]. The existence of these types of pipelines is very essential not only for the improvement of models but also in areas like medicine, remote sensing, and computer vision.

Semantic segmentation is one of the core components of computer vision. It is used in various ways such as analysing remote sensing images, examining medical tissues, and powering self-driving cars[10]. Deep learning (DL) has totally changed the methods of automatic image processing. It allows the usage of data-driven, adaptable, and scalable methods. There are multiple deep architectures in the form of Convolutional Neural Networks (CNNs), Autoencoders, and Generative Adversarial Networks (GANs). They can capture complex changes that make the visual input more realistic, extend it, and compress it. These models take the automation route for the most essential processes from the image domain, like feature extraction, enhancement, and dimensionality reduction. The result is a set of high-quality, standardized inputs that can be further used for more processing[11][12][13]. To take one instance, image classification can only be improved by the use of class activation maps and gradient analysis.

A. Motivation and Contribution

This work was mainly because DL systems need to quickly process and analyses huge volumes of image data. When you manually pre-process such massive datasets, it takes a long time, is simple to make mistakes, and is typically not consistent, which might make the model work worse. This project aims to facilitate data cleaning, augmentation, normalization, and feature extraction through the development of an automated image preparation process. This guarantees that DL models receive standardized, high-quality input. This method is quite useful in the actual world, because there is often a lot of high-dimensional image data accessible. It improves models, making them more powerful, and their ability to distinguish different things gets better. In addition, the load that people and machines have to share is lightened. This research has brought in a variety of significant contributions that are elaborated in detail below:

- Tested the process on a smaller version of the ImageNet dataset with 10 classes. This helped us evaluate its effectiveness. The results showed that it can manage complex, real-world images.
- Cleaned data to fix missing values, remove duplicates and corrupted images, and lower noise.
- Used data augmentation and normalization to increase dataset variety and standardize input values for better model performance.
- Conducted feature extraction to transform raw images into meaningful, compact representations, reducing computational complexity and enhancing model efficiency.
- Designed an Inception-based model tailored for large-scale image classification to effectively learn and extract hierarchical features from preprocessed data.
- Assessed evaluating the model's overall performance using measures such as F1-score, recall, accuracy, and precision.

B. *Organization of the Paper*

The paper is structured as follows: Section II reviews the related work on image preprocessing pipelines, Section III describes the dataset, preprocessing steps, and model implementation, Section IV presents the experimental results with comparative analysis, and Section V concludes the study by highlighting important discoveries and suggesting avenues for further study.

II. Literature Review

A comprehensive review of existing research on image preprocessing pipelines was conducted to support this study. Table I summarizes recent works, outlining the proposed models, datasets, key outcomes, and identified challenges.

Kumar, Singh and Kumar Dewangan (2025) introduces a novel convolutional neural network (CNN) architecture tailored for cactus identification from aerial photographs. The proposed cloud-based pipeline enhances training efficiency through scalable data storage, preprocessing, and distributed training across platforms such as Amazon Web Services (AWS), Google Cloud Platform (GCP), and Microsoft Azure. the model’s computational efficiency, shorter training durations, and cost-effectiveness. The model integrates residual connections and depth wise separable convolutions, achieving 96.7% accuracy on the aerial cactus identification dataset. The results highlight the model’s high performance, cost-efficiency, and scalability, making it suitable for real-world aerial image classification tasks[14].

S and C (2025) a comprehensive analysis of various preprocessing techniques for chromosome metaphase image detection using DL models. Through extensive experimentation, the combination of CLAHE, Gamma Correction, and Median Filtering is shown to outperform other preprocessing methods based on both subjective and objective evaluation metrics. The proposed method achieves exceptional results with an MSE of 3.49, a PSNR of 32.02, an SSIM of 0.99, an Entropy of 4.32, and an Edge Density of 0.03. Moreover, post-detection accuracy is significantly improved, increasing from 73% to 95% when the proposed preprocessing approach is applied before the detection task[15].

Tariku *et al.* (2024), A multi-step pipeline that includes Enhanced Super-Resolution Generative Adversarial Networks (ESRGAN) for resolution augmentation, Contrast-Limited Adaptive Histogram Equalisation (CLAHE) for Image quality is enhanced by contrast enhancement and white balance tweaks for correct colour representation. These preprocessing methods guarantee high-quality input data, which enhances model performance. A remarkable 97.88% accuracy rate was attained by the VGG-16 + SVM model for feature extraction and classification on a dataset[16].

Kussul *et al.* (2024) provides a new way to preprocess Synthetic Aperture Radar (SAR) satellite images to make it easier to find oil spills using DL models. A transfer learning approach with the LinkNet segmentation architecture pre-trained on ImageNet is employed. The model is trained on Sentinel-1 SAR data from 2018–2023 using a designed preprocessing pipeline that converts the single-channel SAR input to a 3-channel RGB image transforming the original SAR intensity values to a normal distribution, extracting nonlinear features, and encoding them into the RGB channels. Quantitative results on a test set show the preprocessed model achieves an improvement of 0.038 in F1-score and 0.054 in Intersection [17].

Sun *et al.* (2023) employing a multi-layer convolutional network to extract the feature information of grape leaves, decreasing model overfitting by adding a data augmentation layer, avoiding aliasing in the gradient direction by using the Leaky Relu activation function, and adding a flat layer By "flattening" convolutional pooling data and one-dimensional multi-dimensional data, the Adam optimizer can increase network efficiency, enhance sparse gradient performance, and shorten training times. The convolutional neural network CNN-GLS model has a 95.13% accuracy rate in identifying tiny samples of grape leaves, according to the experimental data. The model is resilient and has a high capacity for generalization, which is crucial for plant taxonomic identification[18].

Yang *et al.* (2022) proposes YOLO-Xweld, a revolutionary approach that uses the properties of pipeline weld X-ray images to weaken large-scale network detection branches and identify pipeline welding flaws for limited application situations. In order to create a lightweight model and lessen reliance on high-performance hardware, Depthwise separable convolution is implemented. Experiments reveal that the model's size has decreased and the floating-point operations and parameter counts have dropped by almost 70%. In the limited pipeline weld detection context, this is crucial for decreasing reliance on high-performance hardware and improving operability and ease during inspections[19].

Sarkar *et al.* (2022) emphasize melanoma detection in either dermoscopic or non-dermoscopic images, but not both, and offer a unique generalized framework that can identify melanoma in both images. A VGG-16 model is followed by a preprocessing pipeline, data augmentation, and class imbalance resolution. The sensitivity of the model is 87% on non-dermoscopic images and 91% on dermoscopic images (for melanoma patients)[20].

TABLE I. RECENT STUDIES ON IMAGE PREPROCESSING PIPELINES USING DEEP LEARNING

Author & Year	Dataset	Methodology	Key Findings	Limitations / Future Work
Kumar, Singh & Dewangan (2025)	Aerial Cactus Identification Dataset	Proposed a novel CNN architecture with residual connections and depthwise separable convolutions integrated into a cloud-based pipeline (AWS, GCP, Azure) for scalable training and preprocessing.	Achieved 96.7% accuracy; demonstrated high computational efficiency, shorter training times, and cost-effectiveness for aerial image classification.	Future work may focus on optimizing cross-cloud resource allocation and extending model adaptability to other remote-sensing datasets.
S. & C. (2025)	Chromosome Metaphase Image Dataset	Combined CLAHE, Gamma Correction, and Median Filtering as preprocessing techniques before deep learning-based detection.	Improved detection accuracy from 73% to 95%, with performance metrics: MSE=3.49, PSNR=32.02, SSIM=0.99, Entropy=4.32, Edge Density=0.03.	Requires validation on larger, more diverse chromosome datasets; computational complexity of combined preprocessing may be optimized.
Tariku et al. (2024)	Custom Image Dataset	Multi-step preprocessing using ESRGAN for super-resolution, CLAHE for contrast, and white balance adjustment; classification with VGG-16 + SVM.	Achieved 97.88% accuracy; significantly enhanced image quality and feature extraction effectiveness.	Future work could integrate real-time preprocessing and compare with newer hybrid CNN architectures.
Kussul et al. (2024)	Sentinel-1 SAR Satellite Imagery (2018–2023)	Proposed preprocessing converting single-channel SAR to 3-channel RGB via transformation and feature encoding; LinkNet with transfer learning on ImageNet.	Improved F1-score by 0.038 and IoU by 0.054 for oil spill detection; enhanced SAR interpretability.	Model performance needs validation on other satellite sensors; future work may explore real-time detection and multi-temporal analysis.
Sun et al. (2023)	Grape Leaf Image Dataset	Developed CNN-GLS model with data augmentation layer, Leaky ReLU activation, Flatten layer, and Adam optimizer for efficient training.	Achieved 95.13% accuracy with strong robustness and generalization in small-sample grape leaf species recognition.	May extend to larger agricultural datasets and incorporate transfer learning for improved scalability.
Yang et al. (2022)	Pipeline Weld X-ray Image Dataset	Proposed YOLO-Xweld with depthwise separable convolution to detect weld defects; optimized for lightweight operation and reduced computation.	Reduced model parameters and FLOPs by over 70%, maintaining high accuracy; suitable for resource-constrained environments.	Future work includes real-time deployment and further miniaturization for embedded systems.
Sarkar et al. (2022)	Dermoscopic & Non-Dermoscopic Skin Image Datasets	Generalized melanoma detection framework using preprocessing, data augmentation, and VGG-16 for classification.	Achieved 87% sensitivity (non-dermoscopic) and 91% (dermoscopic); improved cross-domain adaptability.	Can be extended to other skin diseases; future research may employ explainable AI for interpretability.

Research gaps: The reviewed studies reveal several research gaps requiring further exploration. While numerous models in areas such as BCIs, medical imaging, agriculture, and industrial inspection perform effectively, issues of generalization, scalability, and computing costs still persist. A lot of study focuses on accuracy but don't look at other important things. These include the ability to understand, resist noise, and work with varied datasets. Preprocessing pipelines generally need to be changed for each dataset, which makes them less useful in diverse sectors. Lightweight and cloud-based solutions have promise, but they also need to think about how much they will cost, how well they will work in real time, and how well they work with other platforms. Future research should aim to create DL pipelines that are resource-efficient, interpretable, and useful in various large-scale and multimodal environment.

III. RESEARCH METHODOLOGY

The Image Preprocessing Pipelines approach uses a smaller version of the ImageNet dataset has 10 classes. After scaling it to 224×224 using bilinear interpolation, it splits it into training, validation, and test sets. It divides it into training, validation, and test sets after applying bilinear interpolation to scale it to 224×224. Data preparation includes handling missing values, eliminating duplicates and distorted, denoising, data augmentation, and min-max normalization in order to maximize a classifier's performance. Feature extraction is used to turn raw Images into a small collection of useful features. This makes the model more efficient and less complicated to compute. The dataset is split in half, with one group getting 80% of the data and the other getting 20%. This keeps the class distribution the same. A CNN-based model is proposed, and its performance is evaluated using confusion matrices to provide a thorough evaluation of categorization results by computing accuracy, precision, recall, and F1-score. Heatmaps and bar charts are examples of data visualizations that are used to examine feature significance and model sensitivity, and the influence of different preprocessing and model parameters. Fig. 1 illustrates the proposed flowchart for Image Preprocessing Pipelines using DL.

The following section presents a thorough breakdown of every stage in the suggested technique:

A. Data Gathering and Analysis

This study utilizes the ImageNet dataset. A subset of the ImageNet dataset with 10 classes that contains large-scale, detailed backgrounds, introducing additional complexity. employ a "320 px" version of it containing approximately 9,500 training images and about 3,920 validation images, from which create a test set by taking 20%, the bilinear interpolation method is used to resize the images to 224×224. Data visualizations such as bar plots and heatmaps were used to examine distribution, feature correlations etc., are given below:

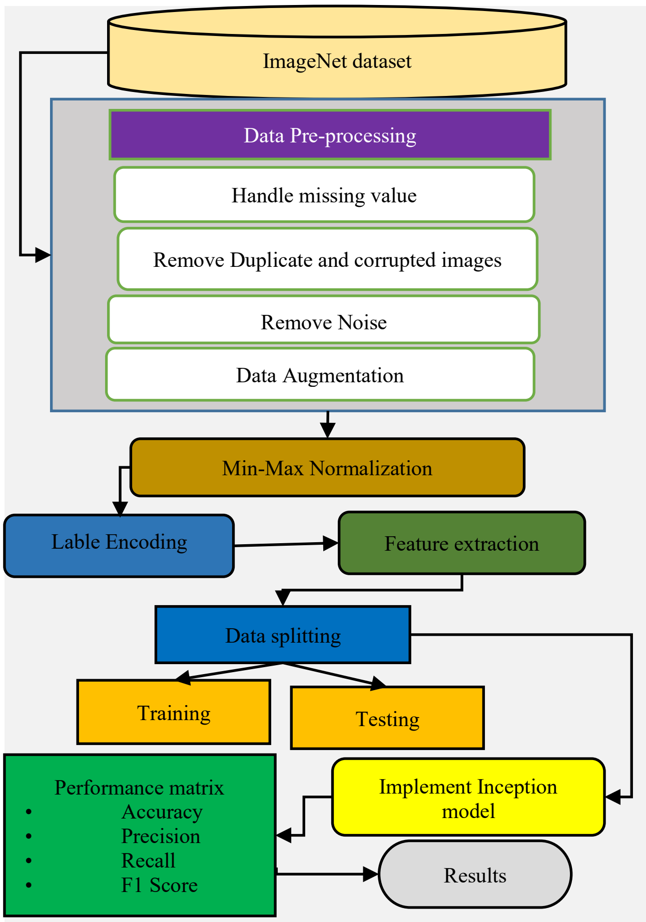

Fig. 1. Proposed flowchart for Image Preprocessing Pipelines using Deep Learning

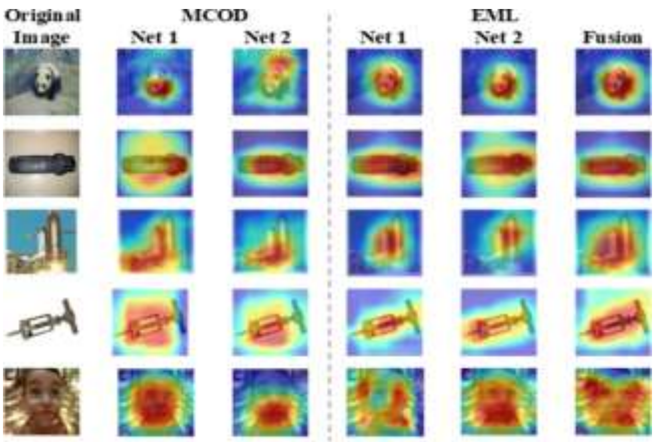

Fig. 2. GRAD-CAM visualization result on the ImageNet dataset

Figure 2 presents Gradient-weighted Class Activation Mapping (GRAD-CAM) visualization results on the ImageNet dataset, comparing MCOD and EMIL saliency detection methods across five sample images (portrait, car, space shuttle, helicopter, face). The figure displays original images alongside saliency maps from MCOD (Net 1, Net 2) and EMIL (Net 1, Net 2, Fusion). Heat maps use thermal coloring where blue denotes low-activation areas and red denotes high-activation areas. Results demonstrate that EMIL's fusion approach achieves superior salient object validating the effectiveness of the proposed ensemble strategy on ImageNet classification tasks.

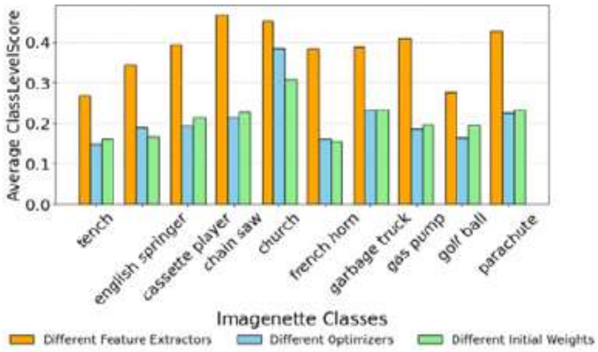

Fig. 3. Bar chart on the ImageNet dataset for Image Preprocessing Pipelines

This figure 3 bar chart presents ImageNet classification performance across ten image classes (beach, English Springer, etc.). Three experimental conditions are compared: different feature extractors (orange bars, ~0.25-0.5 score), different optimizers (blue bars, ~0.15-0.35 score), and different initial weights (green bars, ~0.15-0.25 score). Variations in feature extractors had the greatest effect on classification accuracy for all evaluated classes.

B. Image Pre-processing

The ImageNet dataset was used to get the data ready, which included cleaning, combining, and adding new features. The preparation procedures included dealing with missing values, getting rid of duplicate and damaged Images, getting rid of noise, and then adding data and normalizing it. The main steps in preprocessing are as follows:

- **Handle missing value:** To handle missing values, can delete rows or columns, use imputation to fill in missing values with the mean, median, mode, or a predicted value, or use a data analysis method robust to missing data. The best approach depends on the amount of missing data, its pattern, the type of data (numerical or categorical), and specific analytical goals.
- **Remove Duplicate and corrupted images:** To remove duplicate images, use a duplicate finder app for the operating system by adding folders to scan, then review the results and select files to delete, often by choosing to "Keep All Newest" or similar options.
- **Remove Noise:** Noise reduction in image processing is the process of computationally removing unwanted random variations in pixel intensity (noise) from a digital image to improve its overall clarity and visual quality.
- **Data Augmentation:** The process of making altered copies of preexisting data in order to artificially increase its magnitude is referred to as data augmentation. The main goal of this method is to increase the amount and variety of training datasets in ML, hence enhancing model generalization, mitigating overfitting, and resolving challenges such as class imbalance.

C. Data Encoding

In the context of ML, data encoding is the conversion of data across formats, mainly to allow ML algorithms to use it efficiently. This is particularly crucial for handling categorical data, which consists of labels or categories rather than numerical values. Most ML algorithms are designed to work with numerical input and cannot directly process string-based categorical features.

D. Min-Max Normalization

To normalize the data, the min-max method limited the values to a range of 0 to 1. This was done in an effort to enhance the classifiers' performance and decrease the influence of outliers. The following mathematical formula was used to guide the normalization process (1):

$$X' = \frac{X - X_{min}}{X_{max} - X_{min}} \quad (1)$$

where X is the feature's original value, X' is the normalised value, X_{min} is the feature's minimum value, and X_{max} is the feature's highest value.

E. Feature Extraction

The process of converting unprocessed, high-dimensional data into a smaller number of useful characteristics that keep important information for an ML model. It is a key aspect of data preparation that simplifies complicated datasets, reduces computing costs, prevents overfitting, and ultimately accelerates and enhances algorithms by providing shorter, more meaningful input. In image processing, feature extraction is transforming unprocessed image data into a condensed set of descriptive characteristics that retain significant information while eliminating irrelevant components.

F. Data Splitting

They separated the dataset uses a stratified 80:20 distribution across training and testing sets, ensuring that the class distribution in both selections stayed the same as it was in the original dataset.

G. Proposed Inception Model on image net dataset

The **Inception model**, also known as **Google Net**, is a deep CNN, performs image classification by transforming an input image $x \in R^{H \times W \times D}$ into a high-dimensional feature representation through a hierarchy of convolutional, pooling, and inception modules[21]. The final prediction layer outputs a probability distribution over C possible classes. The classification process can be formulated as eqn (2):

$$\hat{y} = arg \max_{c \in C} P(y = c \mid x; \theta) \quad (2)$$

Where x represents the input image, y is the true class label, θ denotes the learnable parameters of the Inception network, P$(y = c \mid x; \theta)$is the posterior probability of class c, estimated by the network's SoftMax output layer. The SoftMax function converts the raw output logits $z = f_{inception}(x; \theta)$ into normalized probabilities as follows, eqn (3):

$$P(y = c \mid x; \theta) = \frac{e^{z_c}}{\sum_{i=1}^{C} e^{z_i}} \quad (3)$$

where z_c denotes the activation corresponding to class c to optimize the classification performance, the Inception model minimizes the **categorical cross-entropy loss**, defined as eqn (4):

$$L(\theta) = -\sum_{c=1}^{C} y_c log P(y = c \mid x; \theta) \quad (4)$$

where y_c is the one-hot encoded ground-truth label. During training, stochastic gradient descent (SGD) or its variants are used to update the parameters θ, enabling the network.

H. Evaluation metrics

The suggested design's performance was assessed using a variety of measures[22]. A confusion matrix was created to illustrate the categorization findings by displaying the number of correct and incorrect guesses for each class. They got the values for True Positives (TP), False Positives (FP), True Negatives (TN), and False Negatives (FN) from this matrix. Then, as detailed below, these statistics were used to find key performance measures, including accuracy, precision, recall, and F1-score:

Accuracy: The trained model's successful predictions are counted and then divided by the total number of times it received the dataset, which includes input samples. This number is (5)-

$$Accuracy = \frac{\text{TP+TN}}{\text{TP+Fp+TN+FN}} \quad (5)$$

Precision: Precision is calculated by dividing the number of genuine positive predictions by the total number of positive predictions produced by the model. How good the classifier is in predicting the positive classes is expressed as (6)-

$$Precision = \frac{\text{TP}}{\text{TP+FP}} \quad (6)$$

Recall: This statistic represents the percentage of occurrences that were correctly anticipated to be positive to all events that should have been positive. In mathematical form, it is given as (7)-

$$Recall = \frac{\text{TP}}{TP+FN} \quad (7)$$

F1 score: It is the harmonic mean of recall and accuracy, that is, it helps to balance recall and precision. Its range is [0, 1]. Mathematically, it is given as (8)-

$$F1 - score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (8)$$

IV. Results And Discussion

This section shows how the experiment was set up and the model's performance during testing and training. Experiments were implemented in Python running on Google Colab's cloud computing infrastructure with the TensorFlow framework in Jupyter Notebook. GPU acceleration was utilized with dynamically allocated resources: There are 78.2 GB of disk space, 15 GB of GPU, and 12.7 GB of RAM. trained the proposed model using the ImageNet dataset and used the primary metrics in Table II to test it. The Inception model has a 98.7% accuracy rate, a 98.5% precision rate, a 99.5% recall rate, and an F1-score of 98.6%. These findings show that they do a fantastic job of reducing false positives while obtaining a high volume of real positives. The balanced F1-score shows that the model is good and dependable. This illustrates that the automated preprocessing pipeline greatly improves the speed and accuracy of classification for vast amounts of imagine data.

TABLE II. Classification results of the proposed model, Image Preprocessing Pipelines using the ImageNet dataset

Matrix	Inception Model
Accuracy	98.7
Precision	98.5
Recall	99.5
F1-score	98.6

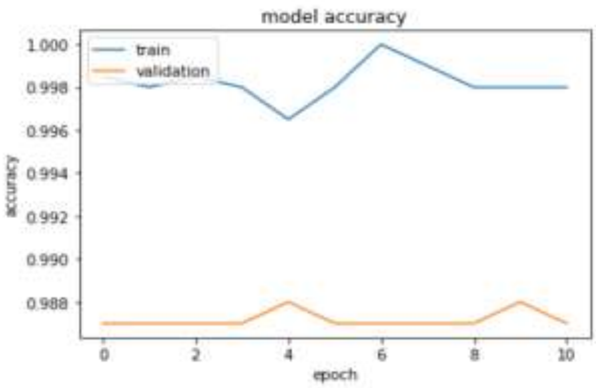

Fig. 4. Accuracy curve for the Inception Model

The Inception model's training and validation accuracy in automated preprocessing workflows. The training accuracy (blue) is almost 100% as epoch 6 concludes, and the validation accuracy (orange) stays at 98.8%. This big difference means that the model is overfitting, which means that it needs better regularization procedures.

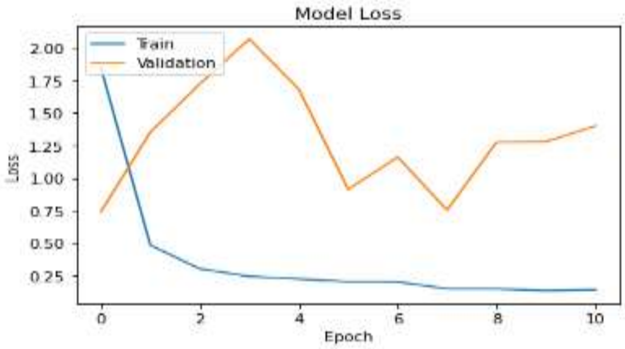

Fig. 5. Loss curve for the Inception model

The loss curves for training and validation when the Inception model was being trained to automatically analyze images. The loss that is recorded during the training (blue) is gradually going down from roughly 1.75 to 0.2, which is an indication that the model is learning. Nevertheless, the validation loss (orange) at epoch 2 goes up sharply and reaches a maximum of 2.0. This suggests that the model is overfitting. This discrepancy between losses indicates that we require more data or some regularization techniques in the data preparation steps.

A. Comparative analysis

Table III is a comparison of the proposed Inception model's performance with that of the existing models. The table here is evaluating the effectiveness of each model. Various deep learning architectures for automated image preprocessing were benchmarked on the ImageNet dataset, and the analysis clearly shows that the Inception model is the most successful one. Among the models compared, VGG16 had the lowest accuracy at 89.71%. On the other hand, ResNet50 was able to achieve a higher accuracy of 93%. DenseNet improved the results significantly by reaching an accuracy of 97.63%. The proposed Inception model, therefore, is more efficient in capturing and processing visual data with the highest accuracy of 98.7%. When compared to other top models, our findings demonstrate that the Inception-based preprocessing pipeline produces stronger and more reliable classification performance.

TABLE III. COMPARISON OF DIFFERENT MACHINE LEARNING AND DEEP LEARNING MODELS FOR IMAGE PREPROCESSING PIPELINES ON THE IMAGENET DATASET

Model	Accuracy
ResNet50 [23]	93
VGC16[16]	89.71
Dense Net[24]	97.63
Inception	**98.7**

The suggested Inception model, with an accuracy of 98.7%, offers numerous benefits for image preprocessing processes on the ImageNet dataset.

V. CONCLUSION AND FUTURE WORK

This paper demonstrates that a variant of the Inception model is effective in the context of systematic image preparation protocols to classify very large volumes of images on the ImageNet dataset. These pipelines also reduce processing power and human interaction required, thus large-scale image analysis is more scalable and reliable. The results reveal that the proposed Automated Image Preprocessing Pipeline significantly enhances the work of the models on large data. Among all the tested architectures, the Inception model with its accuracy of 98.7% has the highest accuracy and is even better than DenseNet (97.63%) VGC16 (89.71%) and ResNet50 (93), which proves that proper preprocessing and the correct choice of feature extraction and model design can contribute significantly to the final classification. More analysis, including testing on an external dataset needs to be done to guarantee that the gains are not in part because of overfitting. The next step will be to apply the suggested framework to the whole ImageNet dataset, which has 1000 classes, to see how well it works for scalability and generalization. Research in the field of automated hyperparameter optimization, introduction of attention models, and creation of lightweight variants of the models that can be deployed to the edges will be undertaken. Also, the cross-dataset validation with medical imaging, autonomous systems and satellite imaging will evaluate the framework's adaptability to many real-world applications.

REFERENCES

[1] C. Shorten and T. M. Khoshgoftaar, “A survey on Image Data Augmentation for Deep Learning,” *J. Big Data*, 2019, doi: 10.1186/s40537-019-0197-0.

[2] Z. Yang, R. O. Sinnott, J. Bailey, and Q. Ke, “A survey of automated data augmentation algorithms for deep learning-based image classification tasks,” *Knowl. Inf. Syst.*, 2023, doi: 10.1007/s10115-023-01853-2.

[3] N. E. Khalifa, M. Loey, and S. Mirjalili, “A comprehensive survey of recent trends in deep learning for digital images augmentation,” *Artif. Intell. Rev.*, 2022, doi: 10.1007/s10462-021-10066-4.

[4] S. Yin *et al.*, “Left Ventricle Contouring in Cardiac Images in the Internet of Medical Things via Deep Reinforcement Learning,” *IEEE Internet Things J.*, 2023, doi: 10.1109/JIOT.2023.3280743.

[5] J. J. Chen *et al.*, “How to do Deep Learning on Graphs with Graph Convolutional Networks,” *IEEE Access*, 2019.

[6] S. D. Verifier and A. H. Drive, “Simulink ® Verification and Validation ™ Reference,” *ReVision*, 2015.

[7] C. S. Senthilnathan Chidambaranathan, Dr Jeyaramalingam Senthil Kumar, Maheshkumar Landge, Dr Mohammed Abdul Salam, Dr Patil Harshal, Srinivasan Rajavelu, Sagili Sreenivas Reddy, “Quantum Machine Learning for AI-Enhanced Image Recognition System,” 2024.

[8] S. R. Sagili, C. Goswami, V. C. Bharathi, S. Ananthi, K. Rani, and R. Sathya, “Identification of Diabetic Retinopathy by Transfer Learning Based Retinal Images,” in *2024 9th International Conference on Communication and Electronics Systems (ICCES)*, IEEE, Dec. 2024, pp. 1149–1154. doi:

10.1109/ICCES63552.2024.10859381.

[9] A. Mumuni and F. Mumuni, "Automated data processing and feature engineering for deep learning and big data applications: A survey," *J. Inf. Intell.*, 2025, doi: 10.1016/j.jiixd.2024.01.002.

[10] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2015. doi: 10.1007/978-3-319-24574-4_28.

[11] J. A. Livezey and J. I. Glaser, "Deep learning approaches for neural decoding across architectures and recording modalities," 2021. doi: 10.1093/bib/bbaa355.

[12] L. F. R. Taveira *et al.*, "Multi-objective Parameter Auto-tuning for Tissue Image Segmentation Workflows," *J. Digit. Imaging*, 2019, doi: 10.1007/s10278-018-0138-z.

[13] N. Jinsakul, C.-F. Tsai, C.-E. Tsai, and P. Wu, "Enhancement of Deep Learning in Image Classification Performance Using Xception with the Swish Activation Function for Colorectal Polyp Preliminary Screening," *Mathematics*, vol. 7, no. 12, 2019, doi: 10.3390/math7121170.

[14] S. Kumar, S. Singh, and N. Kumar Dewangan, "Leveraging Cloud Resources for Distributed Training of Residual CNNs in Aerial Image Classification," *IEEE Access*, vol. 13, pp. 31131–31139, 2025, doi: 10.1109/ACCESS.2025.3542093.

[15] A. S and G. C, "Impact of Advanced Image Preprocessing Techniques on Automated Chromosome Metaphase Detection with Deep Learning Models," in *2025 2nd International Conference on Trends in Engineering Systems and Technologies (ICTEST)*, 2025, pp. 1–7. doi: 10.1109/ICTEST64710.2025.11042662.

[16] G. Tariku *et al.*, "Advanced Image Preprocessing and Integrated Modeling for UAV Plant Image Classification," *Drones*, vol. 8, no. 11, 2024, doi: 10.3390/drones8110645.

[17] N. Kussul, V. Kuzin, Y. Salii, B. Yailymov, and A. Shelestov, "Single-polarized SAR Image Preprocessing in Scope of Transfer Learning for Oil Spill Detection," in *2024 IEEE 12th International Conference on Intelligent Systems (IS)*, 2024, pp. 1–6. doi: 10.1109/IS61756.2024.10705228.

[18] Y. Sun, B. Tian, C. Ni, X. Wang, C. Fei, and Q. Chen, "Image classification of small sample grape leaves based on deep learning," in *ITOEC 2023 - IEEE 7th Information Technology and Mechatronics Engineering Conference*, 2023. doi: 10.1109/ITOEC57671.2023.10291790.

[19] J. Yang, B. Fu, J. Zeng, and S. Wu, "YOLO-Xweld: Efficiently Detecting Pipeline Welding Defects in X-Ray Images for Constrained Environments," in *Proceedings of the International Joint Conference on Neural Networks*, 2022. doi: 10.1109/IJCNN55064.2022.9892765.

[20] T. Sarkar, A. Khare, M. Parekh, P. Mehta, and A. Bhuva, "Robust deep learning framework for the detection of melanoma in images," in *IBSSC 2022 - IEEE Bombay Section Signature Conference*, 2022. doi: 10.1109/IBSSC56953.2022.10037456.

[21] L. Chen, S. Li, Q. Bai, J. Yang, S. Jiang, and Y. Miao, "Review of image classification algorithms based on convolutional neural networks," 2021. doi: 10.3390/rs13224712.

[22] M. Trigka and E. Dritsas, "A Comprehensive Survey of Deep Learning Approaches in Image Processing," *Sensors*, vol. 25, no. 2, 2025, doi: 10.3390/s25020531.

[23] M. J. Yousif, "Enhancing The Accuracy of Image Classification Using Deep Learning and Preprocessing Methods," *Artif. Intell. Robot. Dev. J.*, 2024, doi: 10.52098/airdj.2023348.

[24] P. Parajuli and T. Racharak, "A Framework for Identifying Underspecification in Image Classification Pipeline Using Post-Hoc Analyzer," *Int. Conf. Pattern Recognit. Appl. Methods*, vol. 1, no. Icpram, pp. 426–436, 2025, doi: 10.5220/0013374200003905.

