*International Journal of Applied Engineering & Technology*

# A DEEP LEARNING APPROACH TO ENHANCED FEATURE EXTRACTION FOR COMPUTER VISION

**Gopichand G, Ilayaraja V, Agoorukasetty Adithya, Konduru AdityaVarma, Pranav B S and Bhuvan T**
School of Computer Science and Engineering, Vellore Institute of Technology, Vellore, Tamil Nadu, India,

## ABSTRACT

*Deep learning has significantly advanced the science of computer vision in recent years by delivering world-class results in numerous tasks which includes picture splitting, segmentation, and object recognition. In this research, we look at how deep learning may be used to extract additional data and characteristics from photos. We evaluate the performance of pre-trained convolutional neural network (CNN) models to conventional feature extraction techniques by employing a variety of designs and optimizers. In addition, multi-scale feature extraction and attention methods are applied to boost the model's effectiveness. Our tests show how deep learning may enhance feature extraction, with applications in a variety of industries including surveillance, driverless cars, and medical imaging.*

*Keywords: Convolutional neural networks (CNN), Deep learning, Feature extraction, Image segmentation, Optimizers.*

## 1. INTRODUCTION

In current era, computer vision is hugely replaced by deep learning, allowing for the creation of complex models that are capable of carrying out numerous tasks which includes picture splitting, segmentation, and object recognition numerous tasks which includes picture splitting, object identification, and image segmentation. One of deep learning's main benefits is its ability to automatically extract features from pictures, a crucial step in many computer vision jobs. In this strategy, we suggest leveraging deep learning approaches to improve feature extraction for computer vision. This method is predicated on the notion that we may extract more useful and discriminative features from pictures by using deeper and more complicated neural network topologies, which will increase performance on a variety of computer vision applications.

Convolutional neural networks (CNNs) are used for feature extraction in the suggested method, transfer learning is used to initialise the CNNs with pre-trained weights, and data augmentation is applied to increase the diversity of the training dataset. On the way to enhance the performance of the CNNs, we also suggest using cutting-edge methods like multi-scale feature extraction and attention mechanisms.

We assess the proposed approach on numerous tasks which includes picture splitting, segmentation, and object recognition and picture classification, comparing its performance to that of traditional feature extraction methods. The suggested method is anticipated to have several computer vision applications, including those in fields like surveillance, autonomous cars, and medical imaging.

Deep learning has been a powerful technique in computer vision over the last several years, allowing the development of intricate models that can perform a number of tasks including image segmentation, object detection, and picture categorization. But feature extraction, an essential step in many computer vision applications, continues to be a difficult challenge. The capacity of traditional feature extraction techniques, such as hand-crafted features, to extract relevant and discriminative features from photos is constrained. As a consequence, more sophisticated methods are required to efficiently extract characteristics from pictures and enhance computer vision jobs.This study's goal is to develop and assess a deep learning-based method for improved feature extraction for computer vision applications.

## 2. BACKGROUND RESEARCH

*Jiang et al. [1]* divided feature extraction methods into four groups: machine learning-based statistical methods, methods based on human expert knowledge, and methods based on the local and global structures of images.

*Ismael, Mohammed, Hefny, and others [2]* suggested a model with many steps, including data collection, pre-processing, visualisation, model development, training, and assessment. This research adopts a similar methodology.

A technique for categorising normal instances from AD cases using a computer-aided brain diagnostic (CABD) system was suggested by Acharya, Fernandes, Weikoh, et al. [4]. The technique extracts characteristics from brain pictures using feature mining techniques including CuT, CWT, EWT, and ST. The technique has a high degree of accuracy and could aid in the early detection of AD.

*S. Patil and S. R. Patil. For secure access and security, respectively, et al. [5]* offered the conventional technique and the biometric approach. The biometric technique is utilised when the primary goal is security, while the conventional approach is employed in applications that need secure access.

*He, Paoletti, Haut, Fang* et al. [6] proposed a system for classifying hyperspectral images using a two-step approach involving dimensionality reduction and feature extraction. The method exploits spatial and spectral information and can potentially enhance the accuracy of hyperspectral image seggregation.

A deep learning-based interactive medical picture segmentation approach was suggested by Wang, Zuluaga, Li, Pratt, et al. [7] that increases segmentation accuracy and decreases user involvement for greater accuracy. Convolutional neural networks (CNNs) are included into the pipeline for segmenting data using bounding boxes and scribbles.

In their technique for classifying brain tumours, Varuna Shree, Kumar, and colleagues [9] included preprocessing, segmentation, region-growing, morphological operations, feature extraction, and a probabilistic neural network (PNN) for classification. The technique was very accurate and may help in the detection and management of brain tumours.

A approach for classifying brain tumours was developed by Gumaei, Hassan, Hassan, et al. [10] utilising a hybrid feature extraction method that incorporates the Gist descriptor with the PCA-NGIST method. The technique has a high degree of accuracy and may help with brain tumour identification and therapy.

## 3. METHODOLOGY

Collect and pre-process the dataset: Collect a dataset of images and annotations (if available) for image segmentation. Pre-process the dataset by resizing, normalizing, and splitting it into training and testing sets (Data Augmentation)

Define the model architecture: Choose a suitable deep learning model architecture, such as U-Net, Mask R-CNN, MobileNetV2 for image segmentation.

A series of pooling and convolutional layers are used to extract information from the input image. Then, up sampling layers are used to increase the feature maps' resolution.

Train the model: Use the pre-processed dataset to train the model using a suitable optimizer and loss function. (Adam optimizer and Sigmoid function)

Use the testing set to assess the performance of trained model's *wrt* precision and intersection over union (IoU).

**Fine-tune the model:** If the model does not perform well, fine-tune the model by adjusting the model architecture, hyperparameters, or by collecting more data.

**Apply the model to new images:** Once the model is trained and fine-tuned, it can be applied to new images for segmentation.

**Post-processing:** Perform any necessary post-processing on the segmented images, such as removing small or disconnected regions.

**Evaluation:** Evaluate the model's performance in comparison to the real world or other models.

**Deployment:** Once the model is performing well, it can be deployed on an edge device or cloud.

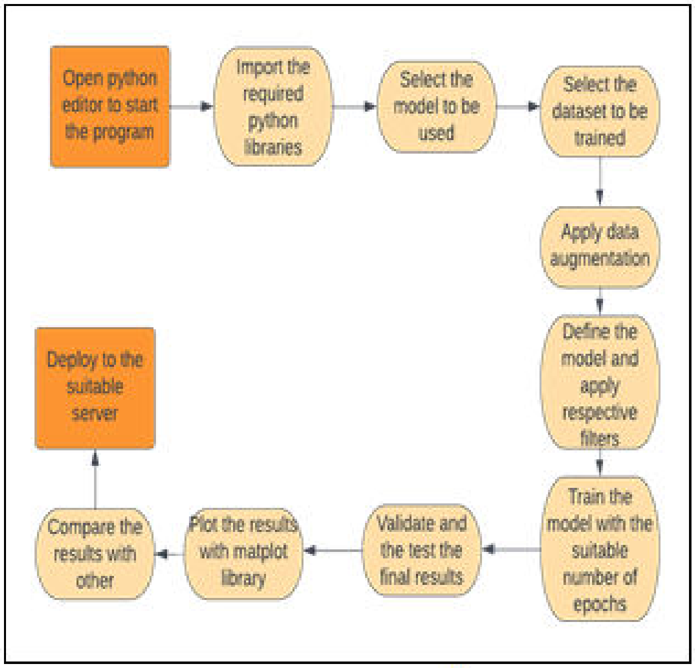


**Figure 1** Process Flow

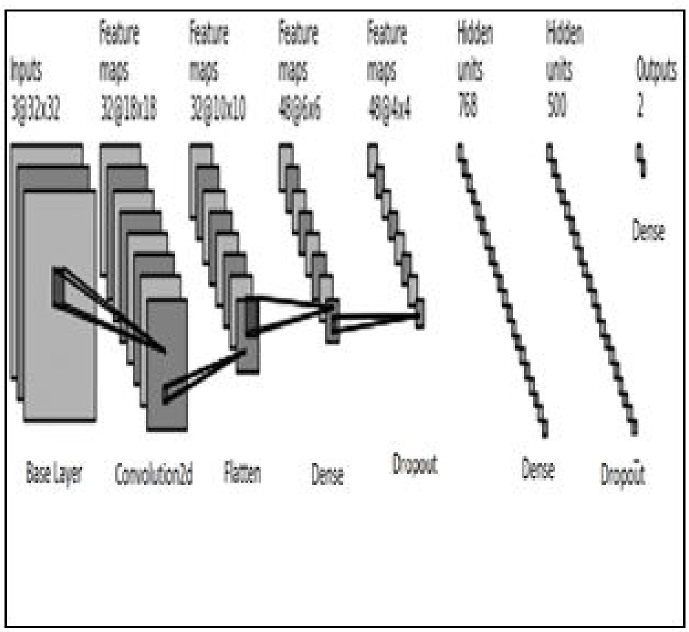


**Figure 2** Representation of Architecture

We use the following pre-trained models for our customization and performance analysis:

1. Resnet50

2. MobileNetV3

3. Resnet_V2

4. Inception_V3

5. Inception_ResentV2 (Hybrid)

### 3.1. Optimizers

**Adam optimizer:**

Artificial neural networks are trained using the optimisation method Adam (Adaptive Moment Estimation). Precisely its a combination of two other optimization algorithms, RMSprop and momentum. The key idea behind Adam is to use the first and second moments of the gradients to update the parameters of the neural network.

$$t = t + 1$$

$$lr\_t = learning\_rate * \frac{sqrt(1 - beta2^{t})}{1 - beta1^{t}}$$

$$mt = beta1 * mt - 1 + (1 - beta1) * gt$$

$$vt = beta2 * vt - 1 + (1 - beta2) * gt^{2}$$

$$wt = wt - 1 - lt\_t * mt / (\sqrt{vt} + \varepsilon)$$

$t$ represents the value of the current parameter, represents the gradient of the current time step, and is the first-order moment respectively and the attenuation coefficient of the second moment; (epsilon) is a small constant used to avoid division by zero. represents the number of time steps, represents the learning rate at the current time step, and represents the first-order moment estimation and second-order moment estimation of the gradient, respectively.

**Stochastic Gradient Descent (SGD):**

Stochastic Gradient Descent (SGD) optimisation algorithm is frequently used in machine learning to update a model's weights as it is being trained. It operates by updating the weights based on a mini-batch, or a small portion of the training data, that is used to compute the gradient of the loss function with respect to the weights.

**Formula:**

$$w = w - alpha * dw$$

$w$ is the weight vector of the model. $alpha$ is the learning rate, which determines the step size taken during the weight update. $dw$ is the gradient of the loss function with respect to the weights, computed on a mini-batch of training data.

## 4. IMPLEMENTATION

**Pre-processing of the data:**

```
train_ds = build_dataset("training")
class_names = tuple(train_ds.class_names)
train_size = train_ds.cardinality().numpy()
train_ds = train_ds.unbatch().batch(BATCH_SIZE)
train_ds = train_ds.repeat()

normalization_layer = tf.keras.layers.Rescaling(1. / 255)
preprocessing_model = tf.keras.Sequential([normalization_layer])

preprocessing_model.add(tf.keras.layers.RandomRotation(40))
preprocessing_model.add(tf.keras.layers.RandomTranslation(0, 0.2))
preprocessing_model.add(tf.keras.layers.RandomTranslation(0.2, 0))
preprocessing_model.add(tf.keras.layers.RandomZoom(0.2, 0.2))
preprocessing_model.add(tf.keras.layers.RandomFlip(mode="horizontal"))


train_ds = train_ds.map(lambda images, labels:
                        (preprocessing_model(images), labels))
```

**Figure 3:** Data augmentation

**Custom Layers while building the model:**

```
model = tf.keras.Sequential([
    # Explicitly define the input shape so the model can be properly
    # loaded by the TFLiteConverter
    tf.keras.layers.InputLayer(input_shape=IMAGE_SIZE + (3,)),
    hub.KerasLayer(model_handle, trainable=do_fine_tuning),
    tf.keras.layers.Flatten(input_shape=input_shape1),
    tf.keras.layers.Dense(256, activation='relu'),
    tf.keras.layers.Dropout(0.6),
    tf.keras.layers.Dense(128, activation='relu'),
    tf.keras.layers.Dropout(rate=0.2),
    tf.keras.layers.Dense(len(class_names),
                          kernel_regularizer=tf.keras.regularizers.l2(0.0001))
])
model.build((None,)+input_shape1)
model.summary()
```

**Figure 4:** Added a combination of custom layers to the pre-trained model

```
Model: "sequential_1"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 keras_layer (KerasLayer)    (None, 2048)              23500352

 flatten (Flatten)           (None, 2048)              0

 dense (Dense)               (None, 256)               524544

 dropout (Dropout)           (None, 256)               0

 dense_1 (Dense)             (None, 128)               32896

 dropout_1 (Dropout)         (None, 128)               0

 dense_2 (Dense)             (None, 5)                 645

=================================================================
Total params: 24,058,437
Trainable params: 558,085
Non-trainable params: 23,500,352
```

**Figure 5:** Summary of the built model

**Performance Metrics:**

Performance metrics of machine learning are used to predict the accuracy and a model's effectiveness. The specific metric used depends on the task and the type of model. Some common performance metrics used in machine learning are:

**Accuracy**: This statistic determines the portion of examples in a dataset that are properly categorised. Although it is often used for classification tasks, it is not necessarily the optimal measure to employ since it may be deceptive when the dataset is imbalanced

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN}$$

**Precision and Recall**: While recall gauges the ratio of positive forecasts accuracy among all original positive occurrences, precision scales the ratio of accurate positive forecasts among all positive forecasts. These metrics are used to evaluate how well categorization algorithms perform.

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

$$IoU = \frac{Object \cap Detected\ box}{Object \cup Detected\ box}$$

**Mean Squared Error (MSE) and Mean Absolute Error (MAE)**: Regression models' performance is assessed using these metrics, where MSE represents the middling squared difference among the predicted and actual units and MAE indicates the typical absolute difference.

$$MSE = \frac{1}{n}\sum(y - \hat{y})^2$$

$$MAE = \frac{1}{n}\sum|y - \hat{y}|$$

## 5. RESULTS AND DISCUSSION

For each pre trained model extracted from the resource, the following steps are performed for training and performance evaluation:

1. Importing the dataset.
2. Performing data augmentation and pre-processing.
3. Building the model with the help of transfer learning and image processing techniques.
4. Compiling and training the built model.
5. Conducting performance analysis

Initially, all the mentioned models were built with Stochastic Gradient Descent (SGD) optimizer and compared. It was observed that Resent50 and MobileNetV3 perform well with small and considerable size datasets. Therefore, further analysis was performed on these two pre-trained models using the Adam optimizer. It was concluded that both the models perform very well with the combination of our image processing layers.

**Table 1:** Metrics using SGD optimizer

| NO. | ARCHITECUTRE | EPOCH | LOSS | ACCURACY | VAL_LOSS | VAL_ACCURACY |
|---|---|---|---|---|---|---|
| 1 | Inception V3 | 50 | 0.6798 | 0.8836 | 0.6229 | 0.8972 |
| 2 | Mobilenet_v3_large_100_224 | 50 | 0.6407 | 0.8966 | 0.5933 | 0.9069 |
| 3 | Resnet_v2_152 | 20 | 0.78 | 0.8257 | 0.6832 | 0.8653 |
| 4 | Inception_resnet_v2 | 30 | 0.7415 | 0.8521 | 0.6739 | 0.8597 |

**Table 2:** Metrics using Adam optimizer

| No. | Architecture | Epoch | Loss | Accuracy | Precision | Recall | MSE | MAE |
|---|---|---|---|---|---|---|---|---|
| 1 | Mobilenet_v3_large_100_224 | 100 | 0.4858 | 0.9688 | 0.6765 | 0.9818 | 2.6410 | 1.2907 |
| 2 | Resnet50 | 100 | 0.4769 | 0.9747 | 0.2250 | 0.9993 | 11.9335 | 2.7894 |

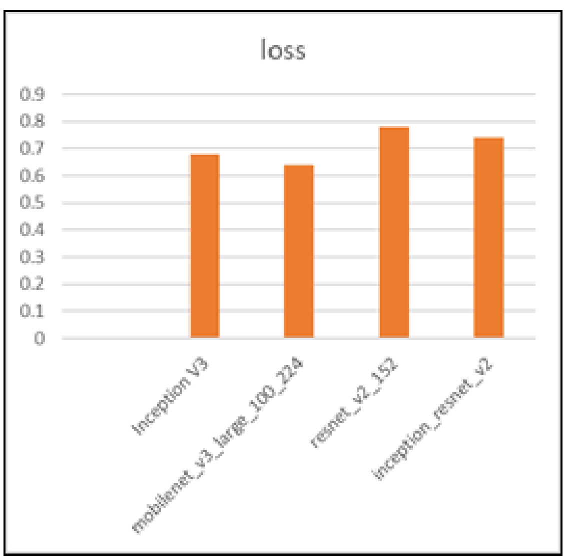


**Figure 6:** Loss with SGD optimizer

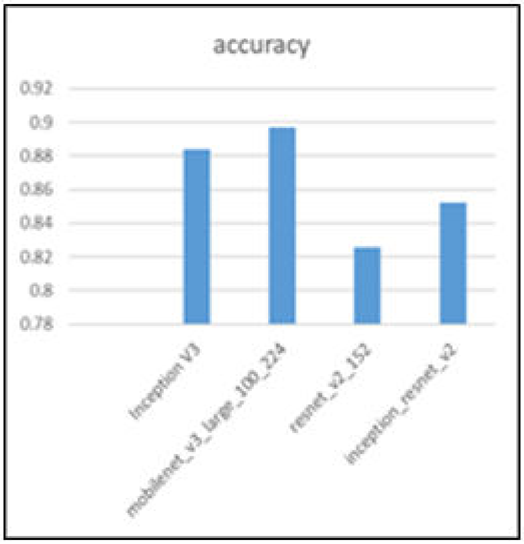


**Figure7:** Accuracy with SGD optimizer

**1. Resnet50 - Using Adam optimizer**

**Performance Evaluation of Resnet50**

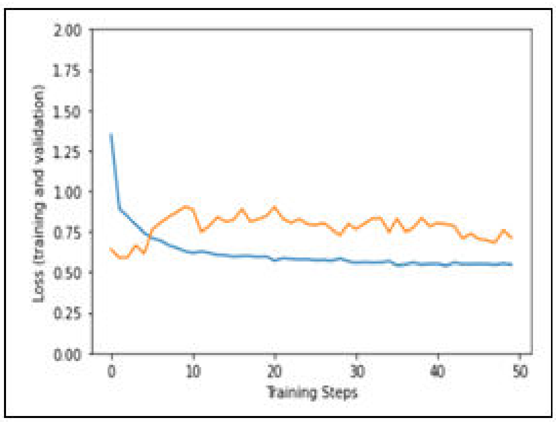


**Figure 8:** Loss metric

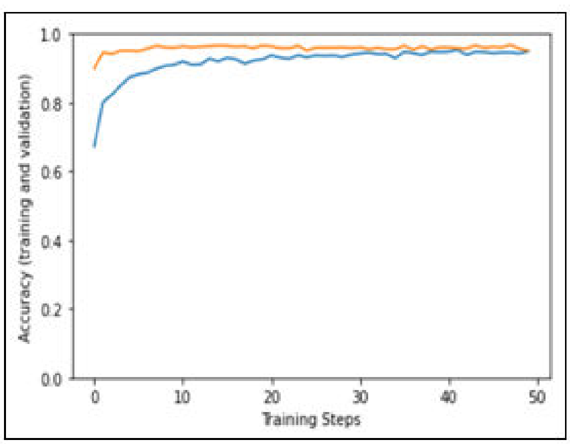


**Figure 9:** Accuracy

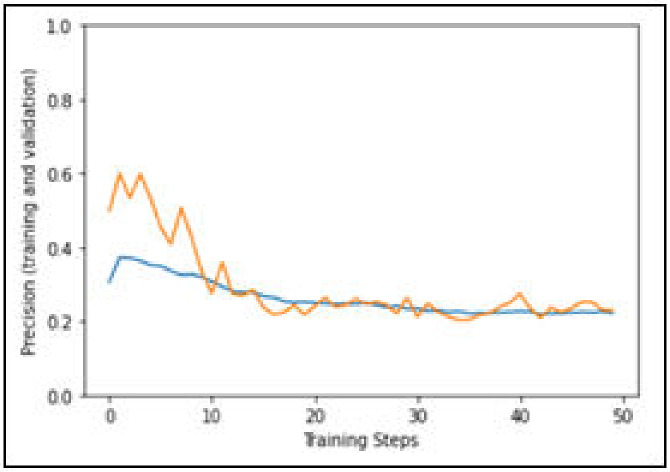


**Figure 10:** Precision

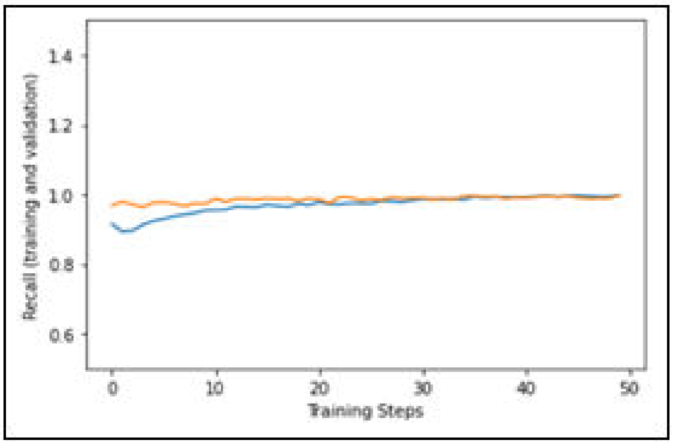


**Figure 11:** Recall

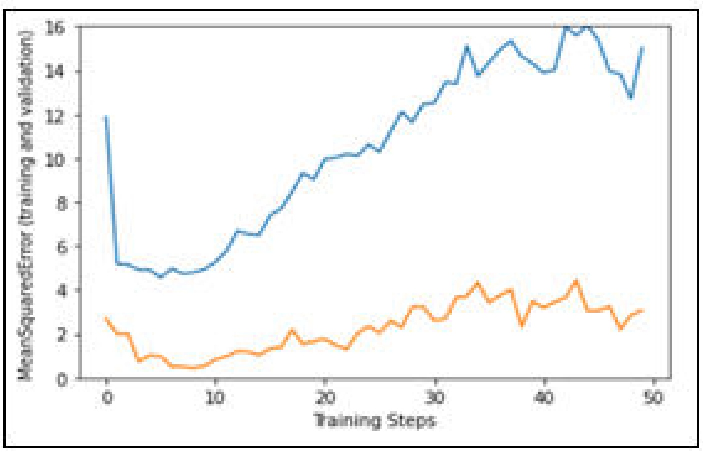


**Figure 12:** Mean Squared Error

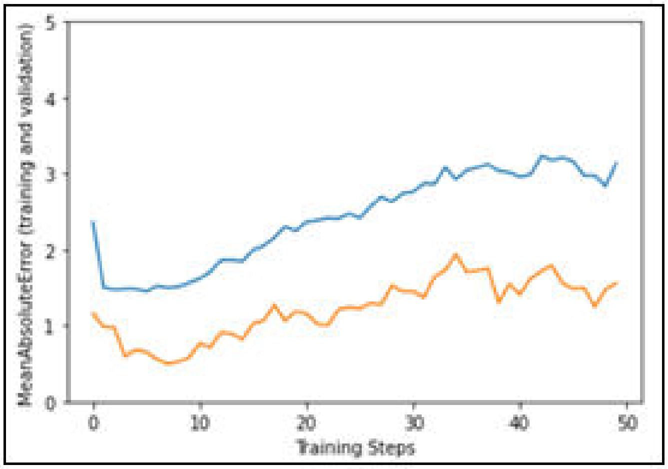


**Figure 13:** Mean absolute Error

**2. MobileNetV3 with Adam optimizer**

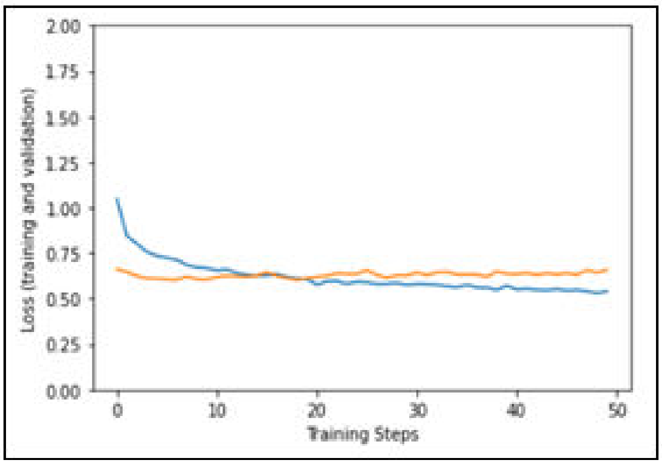


**Figure 14:** Loss metric

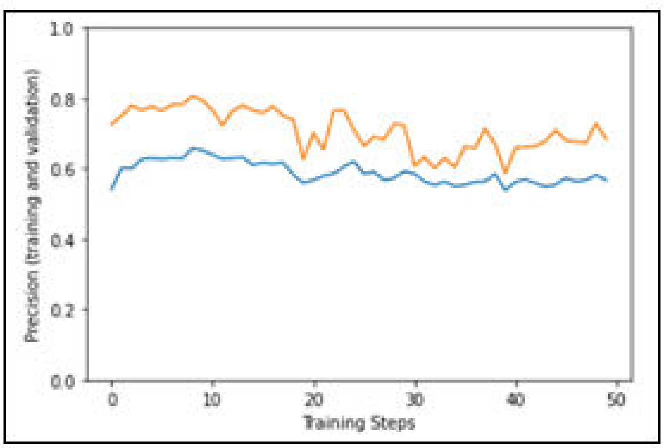


**Figure 15:** Precision

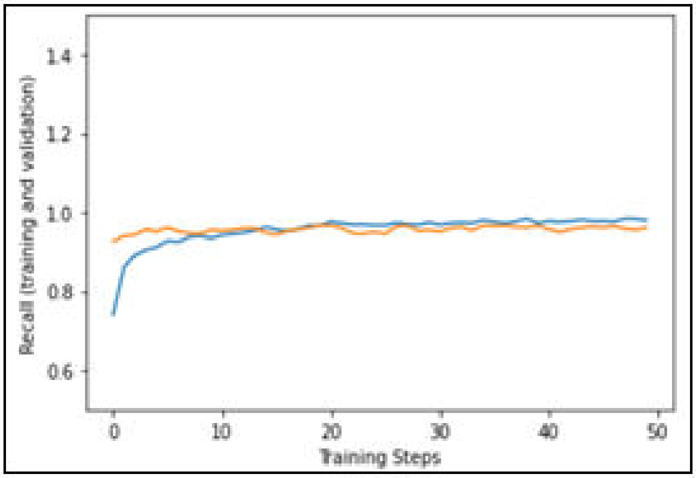


**Figure 16:** Recall

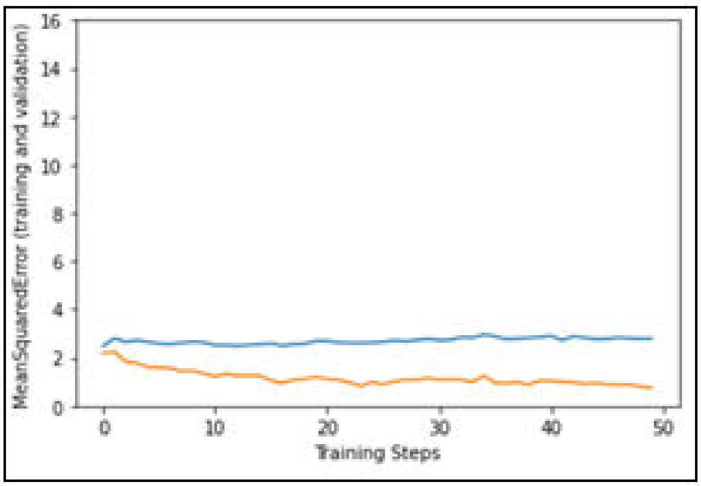


**Figure 17:** Mean Squared Error

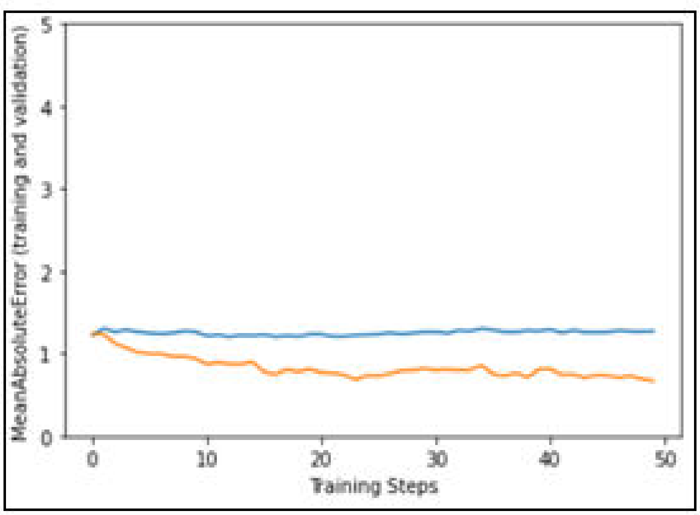


**Figure 18:** Mean Absolute Error

**Accuracy vs Loss graph**

**Resnet50:**

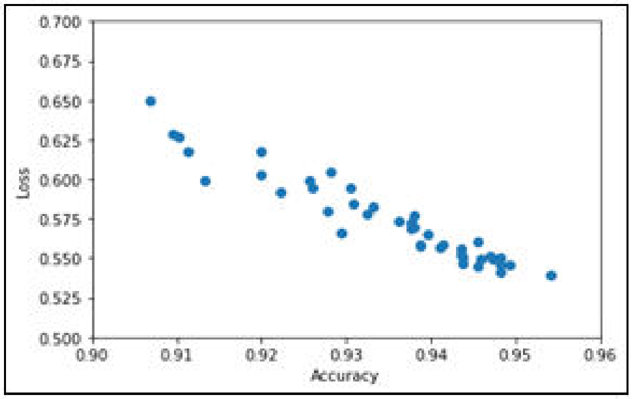


**Figure 19:** Accuracy vs Loss (Resnet50)

**MobileNetV3:**

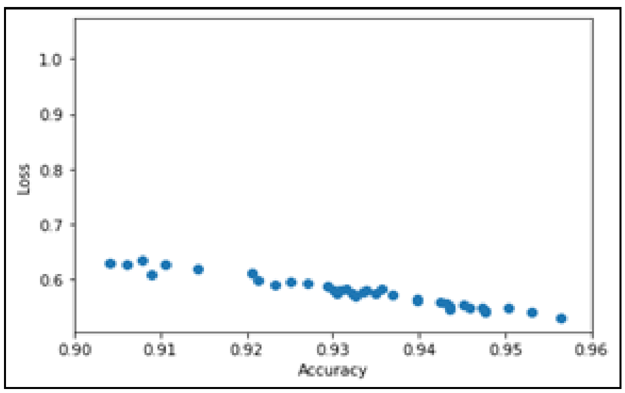


**Figure 20:** Accuracy vs Loss (MobileNetV3)

## 6.CONCLUSION AND FUTURE WORK

After conducting extensive research and testing with various pre-trained models, it has been proven that transfer learning is a highly effective method for improving object detection models, particularly when combined with customized image processing layers. This study reinforces the idea that artificial intelligence models can be easily tailored to meet our specific needs. Moreover, the paper proposes a combination of image processing layers and their corresponding optimizers to enhance the performance of existing pre-trained models.

## REFERENCES

[1] Jiang X. Feature extraction for image recognition and computer vision. In 2009 2nd IEEE international conference on computer science and information technology 2009 Aug 8 (pp. 1-15). IEEE.

[2] Ismael SA, Mohammed A, Hefny H. An enhanced deep learning approach for brain cancer MRI images classification using residual networks. Artificial intelligence in medicine. 2020 Jan 1;102:101779.

[3] Salau AO, Jain S. Feature extraction: a survey of the types, techniques, applications. In 2019 international conference on signal processing and communication (ICSC) 2019 Mar 7 (pp. 158-164). IEEE.

[4] Acharya UR, Fernandes SL, WeiKoh JE, Ciaccio EJ, Fabell MK, Tanik UJ, Rajinikanth V, Yeong CH. Automated detection of Alzheimer's disease using brain MRI images–a study with various feature extraction techniques. Journal of Medical Systems. 2019 Sep;43:1-4.

[5] Patil S, Patil SR. Enhancement of feature extraction in image quality. In 2019 Third International conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud)(I-SMAC) 2019 Dec 12 (pp. 490-495). IEEE.

[6] He N, Paoletti ME, Haut JM, Fang L, Li S, Plaza A, Plaza J. Feature extraction with multiscale covariance maps for hyperspectral image classification. IEEE Transactions on Geoscience and Remote Sensing. 2018 Aug 17;57(2):755-69.

[7] Wang G, Li W, Zuluaga MA, Pratt R, Patel PA, Aertsen M, Doel T, David AL, Deprest J, Ourselin S, Vercauteren T. Interactive medical image segmentation using deep learning with image-specific fine tuning. IEEE transactions on medical imaging. 2018 Jan 26;37(7):1562-73.

[8] Wang G, Zuluaga MA, Li W, Pratt R, Patel PA, Aertsen M, Doel T, David AL, Deprest J, Ourselin S, Vercauteren T. DeepIGeoS: a deep interactive geodesic framework for medical image segmentation. IEEE transactions on pattern analysis and machine intelligence. 2018 Jun 1;41(7):1559-72.

[9] Varuna Shree N, Kumar TN. Identification and classification of brain tumor MRI images with feature extraction using DWT and probabilistic neural network. Brain informatics. 2018 Mar;5(1):23-30..

[10] Gumaei A, Hassan MM, Hassan MR, Alelaiwi A, Fortino G. A hybrid feature extraction method with regularized extreme learning machine for brain tumor classification. IEEE Access. 2019 Mar 10;7:36266-73.

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

[12] Mukti IZ, Biswas D. Transfer learning based plant diseases detection using ResNet50. In 2019 4th International conference on electrical information and communication technology (EICT) 2019 Dec 20 (pp. 1-6). IEEE.

[13] Sharma N, Jain V, Mishra A. An analysis of convolutional neural networks for image classification. Procedia computer science. 2018 Jan 1;132:377-84.

[14] Zhang Z. Improved adam optimizer for deep neural networks. In2018 IEEE/ACM 26th international symposium on quality of service (IWQoS) 2018 Jun 4 (pp. 1-2). Ieee..

[15] Li Z, Liu F, Yang W, Peng S, Zhou J. A survey of convolutional neural networks: analysis, applications, and prospects. IEEE transactions on neural networks and learning systems. 2021 Jun 10.

[16] Albawi S, Mohammed TA, Al-Zawi S. Understanding of a convolutional neural network. In 2017 international conference on engineering and technology (ICET) 2017 Aug 21 (pp. 1-6). Ieee.

[17] Zhao ZQ, Zheng P, Xu ST, Wu X. Object detection with deep learning: A review. IEEE transactions on neural networks and learning systems. 2019 Jan 27;30(11):3212-32.

[18] Shah M, Kapdi R. Object detection using deep neural networks. In 2017 International Conference on Intelligent Computing and Control Systems (ICICCS) 2017 Jun 15 (pp. 787-790). IEEE.

[19] Lu S, Wang B, Wang H, Chen L, Linjian M, Zhang X. A real-time object detection algorithm for video. Computers & Electrical Engineering. 2019 Jul 1;77:398-408.

[20] Chen YP, Li Y, Wang G. An Enhanced Region Proposal Network for object detection using deep learning method. PloS one. 2018 Sep 20;13(9):e0203897.