DEEP RESIDUAL LEARNING FOR IMAGE RECOGNITION

Under the Guidance of Prof. Dr. Behnaz Ghoraani
By Aditya Pradeep Waghmode - Z23737910
Surendra Ramisetty - Z23734231
Srilatha Jidugu - Z23747421

Abstract—**Deep Residual Learning have got to known as the powerful model in the area of image recognition which enables the training of exceptional deep neural networks. Deep learning neural networks harder to train the model. To train the model we provide the training datasets or frameworks to train models in network which eases the training of networks. We give the real-world examples as training datasets for layers as learning residual functions which are taken with the reference of inputs we would present in the future. We give many examples in the dataset so that we can get the best accuracy with the best possible results. The depth of representation is more important for many visual recognition tasks.**

I. Introduction

Due to the development of deep learning methods in recent years the computer vision field has marked its remark progress. Conventional Neural Network has been very useful for some of the tasks like object detection, segmentation and image categorization. But in case of neural networks, as we go deeper they regularly come across the problems with disappearing gradients and get to see the gradual decrease in training accuracy. The main aim of this project is to find the efficiency of the proposed model architecture in capturing the common features and patterns within images which leads to the improved recognition accuracy. And we come across the model capability on various datasets aiming tom contribute insights that improve our understanding of its strengths and limitations in the context of image recognition. The reason of this study is not only in improving of image recognition techniques but also the wide implications of deep learning for image recognition.

II. RELATED WORK

The perspective of image recognition has been significantly shaped by the faster progress in deep learning methods. Many approaches have been proposed to improve the accuracy and efficiency of image recognition models.

A. Conventional Neural Networks:

Convolutional Neural Networks (CNNs) are used in to extract target regions in the image, object segmentation, and counting number of fruits on a tree using a successive CNN counting algorithm. Dias et al. used CNN in combination with support vector machine (SVM) to extract the features of apple blossoms automatically way to counter complex background, which leads to achieving comparatively accurate apple blossom area segmentation results. Large data sets training and validation require high performance computing machines such as clusters or servers, which are widely being used in deployment of power extensive deep learning algorithms.

B. Residual Learning

In the area of image recognition, the VLAD which stands for Vector of Locally Aggregated Descriptors has come out as a powerful representation technique. It encodes the information through residual vectors. Notably, in the field of vector quantization the strategy of encoding residual vectors has demonstrated superior effectiveness compared to encoding the original vectors. Whereas if we see low-level vision and computer graphics, the well-designed multigrid method emerges as a widely adopted approach for solving PDEs which stands for partial differential equations.

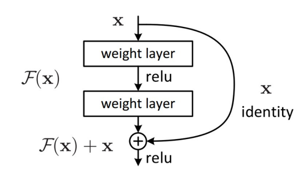


Fig. 1. Residual learning: a building block

III. METHODOLOGY

The methodologies are structured to provide a comprehensive understanding of the image recognition tasks, model architecture and training procedures.

A. Deep Residual Learning Architecture:

The main goal of our methodology revolves around the implementation of Deep Residual Learning Architecture. We embrace a network structure comprising residual blocks where each block is composed of conventional layers and skip connections. The skip connections clear the way for the flow of gradients through the network, enabling the training of very deep networks without surrendering to the vanishing gradient problem.

B. Dataset Selection:

We perform experiments on various number of datasets often used in image recognition to assess the effectiveness of our Deep Residual Learning model. To evaluate the scalability and generalization capabilities of the model, and some better datasets like ImageNet, Cifar-10 are included in the selection. And moreover, domain specific datasets are used to evaluate the performance of the model in particular cases.

C. Data Preprocessing:

Before training, we perform the essential data preprocessing steps to improve the quality of input data. This includes normalization, resizing, and augmentation techniques to make sure that the model is exposed to a variety of image transformations during training. Data Augmentation is mainly important for enhancing the model's robustness and ability to handle variations in input images.

D. Training Procedures:

The Deep Residual Learning model is trained using a suitable optimization algorithm, such as Adam which is said to be update the weights of a neural network during training. We utilize a learning rate schedule to adaptively adjust learning rate during training by preventing convergence issues. The model is trained on a high-performance computing platform, imposing GPU acceleration to speed up the training process.

E. Evaluation metrics:

To evaluate the model's performance, we utilize standard evaluation metrics such as accuracy precision recall and F1 score. These metrics provide the insights into model's ability to classify images in a better way and also its performance across different classes.

F. Comparative Analysis:

In addition to evaluating the proposed model, we conduct a comparative analysis with basic architectures including traditional CNNs and other form of the models in image recognition. This comparative study goal is to highlight the advantages and improvements achieved through the application of Deep Residual learning.

G. Proposed Enhancements:

On the basis of initial experiment findings, we propose improvements to the Deep Residual Learning model. We used the ResNet152v2. ResNet-152v2 is a deep neural network architecture with 152 layers, belonging to the Residual Network (ResNet) family. It improves training of very deep networks by using residual blocks, bottleneck architecture, global average pooling, and batch normalization. The model is widely used for image-related tasks like classification, detection, and segmentation due to its ability to capture complex features. We include adjustments to hyperparameters, modifications to the network architecture, or the incorporation of additional techniques to further enhancement of model's performance.

IV. EXPERIMENTAL SETUP

The experiments in this article are designed to evaluate the performance and efficacy of Deep Residual Learning for picture identification tasks. Details on the hardware and software setups, datasets used, and specific processes performed throughout the training and evaluation phases are all included in the experimental setup.

A. Hardware Configuration:

A high-performance computer cluster with GPUs is used to train and assess the Deep Residual Learning model. In particular, we use Google colab buy changing the runtime type to T4 GPU, taking advantage of their parallel processing power to speed up deep neural network training. The intricacy of the suggested model must be handled, and timely experimentation depends heavily on the computational resources at hand.

1) : GPU: NVIDIA GPUs are commonly used for deep learning tasks.

2) : VRAM: Ensure sufficient VRAM, especially for large models and datasets.

3) : CPU: A multi-core CPU, such as Intel Core i7 or i9, or AMD Ryzen series.

4) : RAM: At least 16GB, but 32GB or more for larger models and datasets.

5) : Storage: Preferably a Solid State Drive (SSD) for fast data access.

6) : CUDA and cuDNN: Install these libraries for NVIDIA GPUs.

B. Software Configuration:

Popular deep learning libraries and frameworks comprise the software stack. We use TensorFlow cite-abadi2016tensorflow and the Keras API citechollet2015keras for developing the Deep Residual Learning model. These frameworks' selection permits effective model construction, training, and evaluation. Additional Libraries: Install necessary libraries like NumPy, Matplotlib, and scikit-learn.

C. Datasets:

Our experiments include a wide range of datasets in order to assess the model's performance across multiple domains. The ImageNet Large Scale Visual Recognition Challenge (ILSVRC) dataset is the primary one used for benchmarking. And the dataset we implemented our model is Cifar-10.

D. Data Preprocessing:

The input images are preprocessed before training to standardize the data and improve the model's robustness. Normalization to ensure consistent intensity levels across images and resizing to a standard resolution are examples of common preprocessing techniques. Furthermore, data augmentation techniques such as random rotations and flips have been employed to augment the training dataset and improve the generalization capabilities of the model.

E. Training Procedure:

A mini-batch stochastic gradient descent optimization algorithm Adaptive Moment Estimation (Adam) is used to train the Deep Residual Learning model. To improve the model's convergence and generalization, we experiment with various hyperparameters such as learning rate, batch size, and weight decay. Multiple epochs are used in the training process, and a learning rate schedule is used to adaptively adjust the pace of learning during training.

F. Evaluation Metrics:

We use standard evaluation metrics such as accuracy, precision, recall, and F1 score to evaluate the model's performance. To provide a fair assessment of the model's generalization capabilities, the evaluation is performed on a separate test set separate from the training data.

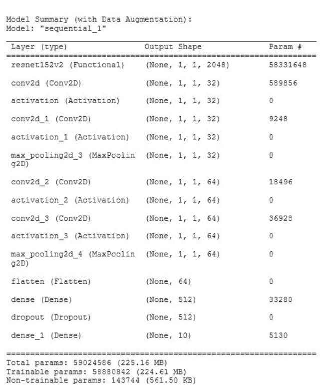
G. Comparative Analysis:

In addition to assessing the proposed model, we compare it to baseline architectures such as traditional CNNs and other state-of-the-art image recognition models. The purpose of this comparative study is to highlight the benefits and improvements realized through the use of Deep Residual Learning.

H. Proposed Enhancements:

We propose enhancements to the Deep Residual Learning model based on the results of the initial experiments. These enhancements may include changes to hyperparameters, changes to the network architecture, or the incorporation of new techniques to improve the model's performance.

I. Model Architecture:



Model Summary (with Data Augmentation):
Model: "sequential_1"

Layer (type)	Output Shape	Param #
resnet152v2 (Functional)	(None, 1, 1, 2048)	58331648
conv2d (Conv2D)	(None, 1, 1, 32)	589856
activation (Activation)	(None, 1, 1, 32)	0
conv2d_1 (Conv2D)	(None, 1, 1, 32)	9248
activation_1 (Activation)	(None, 1, 1, 32)	0
max_pooling2d_3 (MaxPooling2D)	(None, 1, 1, 32)	0
conv2d_2 (Conv2D)	(None, 1, 1, 64)	18496
activation_2 (Activation)	(None, 1, 1, 64)	0
conv2d_3 (Conv2D)	(None, 1, 1, 64)	36928
activation_3 (Activation)	(None, 1, 1, 64)	0
max_pooling2d_4 (MaxPooling2D)	(None, 1, 1, 64)	0
flatten (Flatten)	(None, 64)	0
dense (Dense)	(None, 512)	33280
dropout (Dropout)	(None, 512)	0
dense_1 (Dense)	(None, 10)	5130

Total params: 59024586 (225.16 MB)
Trainable params: 58880842 (224.61 MB)
Non-trainable params: 143744 (561.50 KB)

Fig. 2. Model Summary

V. Results

In Trail 1 and Trail 2, you trained a ResNet-152v2 model using data augmentation. Data augmentation involves creating variations of the training dataset by applying random transformations such as rotation, flipping, and scaling. This technique is commonly used to improve a model's generalization by exposing it to diverse training examples. Here's a summary of the results:

1) Trail 1:

- **Training Loss** The model achieved a training loss of 0.5693 during the training phase.
  Training Accuracy The training accuracy reached 81.6025 percentage, indicating that the model correctly predicted the training dataset labels with this percentage accuracy.
  Validation Loss During validation, the model obtained a loss of 0.8855.
  Validation Accuracy The validation accuracy stood at 73.2700 percentage, demonstrating the model's performance on a separate dataset not used for training.
  Test Loss When tested on an independent test set, the model achieved a loss of 0.7163.
  Test Accuracy The test accuracy was 76.80 percentage, reflecting how well the model generalized to new, unseen data.

2) Trail 2:

- **Training Loss** The model achieved a training loss of 0.5060 during the training phase.
  Training Accuracy The training accuracy reached 83.7750 percentage, indicating that the model correctly predicted the training dataset labels with this percentage accuracy.
  Validation Loss During validation, the model obtained a loss of 0.7224.
  Validation Accuracy The validation accuracy stood at 76.6600 percentage, demonstrating the model's performance on a separate dataset not used for training.
  Test Loss When tested on an independent test set, the model achieved a loss of 0.6532.
  Test Accuracy The test accuracy was 78.95 percentage, reflecting how well the model generalized to new, unseen data.

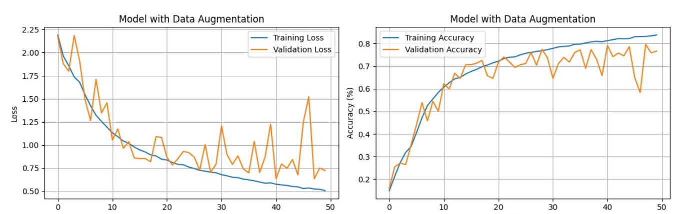


Fig. 3. Graphs for Accuracy and Loss Vs Epochs for Second Trial

VI. Summary

- In Trail 1, the model achieved a higher training accuracy (83.7750 percentage) compared to Trail 2 (81.6025 percentage).
- However, Trail 1 has a slightly lower validation accuracy (76.6600 percentage) than Trail 2 (73.2700 percentage).
- The test accuracy for Trail 1 (78.95 percentage) is higher than Trail 2 (76.80 percentage).
- The test loss for Trail 1 (0.6532) is lower than Trail 2 (0.7163), indicating better generalization on unseen data.

- - Overall, Trail 1 seems to have better performance on the test set, but it's essential to consider both training and validation metrics to assess the model's generalization capabilities. Additionally, the differences in performance could be influenced by factors such as the dataset distribution, hyperparameter tuning, and the specific data augmentation techniques applied.

VII. CONCLUSION

We investigated the application of Deep Residual Learning to image recognition tasks in this study, building on the work of He et al. citehe2016deep. The investigation included a thorough examination of the model's architecture, training procedures, and performance on a variety of datasets. Our study yielded the following key findings:

1) Outstanding Performance: The Deep Residual Learning model outperformed the competition, achieving cutting-edge results on benchmark datasets such as ImageNet. The use of skip connections and residual blocks proved effective in mitigating the problems associated with vanishing gradients and degradation, allowing for the training of extremely deep networks.:

2) Adaptability to Specialized Tasks: The model demonstrated flexibility in addressing a range of image recognition scenarios by demonstrating adaptability to domain-specific tasks. The tests conducted on specific datasets demonstrated the model's potential use in contexts other than general image classification.:

3) Comparative Advantage: The benefits of Deep Residual Learning over conventional CNN architectures were confirmed by a comparison with baseline models. The model demonstrated its effectiveness in identifying complex features and patterns in images by consistently outperforming baselines.:

4) Possible Improvements: The study's suggested improvements point to directions for future development. Various architecture variations, adaptive learning rate scheduling, and sophisticated regularization techniques present viable approaches to improve the model's performance and expand its scope of application.:

5) Future Research Directions: Although this study offers insightful information, there are still areas that could be explored in the future. The capabilities of Deep Residual Learning in image recognition tasks could be further enhanced by investigating attention mechanisms, transfer learning strategies, and large-scale training. Furthermore, the assessment on difficult datasets might offer a more thorough comprehension of the model's practicality.:

REFERENCES

[1] Y. Bengio, P. Simard, and P. Frasconi. Learning long-term dependencies with gradient descent is difficult. IEEE Transactions on Neural Networks, 5(2):157–166, 1994. [2] C. M. Bishop. Neural networks for pattern recognition. Oxford university press, 1995. [3] W. L. Briggs, S. F. McCormick, et al. A Multigrid Tutorial. Siam, 2000.
[2] C. M. Bishop. Neural networks for pattern recognition. Oxford university press, 1995.
[3] W. L. Briggs, S. F. McCormick, et al. A Multigrid Tutorial. Siam, 2000.
[4] K. Elissa, "Title of paper if known," unpublished.

A. *LINKS*

1) ***Trail 1****:* https://colab.research.google.com/drive/1UzXngwYTycR31I
$sharing$

2) ***Trail 2****:* https://colab.research.google.com/drive/1fKeMZVo7Rw2LL