Deeper Networks for Image Classification

Lavanya Kebbhalli Muruli
ECS795P – Deep Learning and Computer Vision
School of Electronic Engineering and Computer Science - Department of Computer Science
M.Sc. Artificial Intelligence
Queen Mary University
London, United Kingdom
l.kebbhallimuruli@se23.qmul.ac.uk
230383359

Abstract—This paper represents a comparison between different Convolutional Neural Networks. In particular, the models GoogLeNet, and VGG are compared on different datasets, such as MNIST and Cifar10.

I. INTRODUCTION

This paper represents a comparison between different Convolutional Neural Networks. Handwritten digit recognition is a fundamental task in computer vision with various applications, including digitizing documents and postal automation. Deep learning models have shown remarkable performance in this task, prompting the exploration of different architectures to improve accuracy and efficiency. In particular, the models GoogLeNet and VGG. These two models are compared on the datasets MNIST and Cifar10.

II. LITERATURE REVIEW

A. Neural Network models

1) GoogLeNet: GoogleNet, also known as Inception-v1, revolutionized convolutional neural network (CNN) architectures with its inception modules, introduced by Szegedy et al. in 2014. These modules utilize parallel convolutional layers of varying kernel sizes to efficiently capture features at multiple scales, reducing computational complexity while maintaining accuracy. The network's deep and wide structure enables it to learn diverse feature representations within the same layer, enhancing its ability to extract hierarchical features from visual data. Moreover, GoogleNet introduced auxiliary classifiers to address the vanishing gradient problem, aiding in convergence and regularization during training. Its success has inspired subsequent iterations of the Inception architecture and established its effectiveness across a range of computer vision tasks, including classification, object detection, and segmentation.

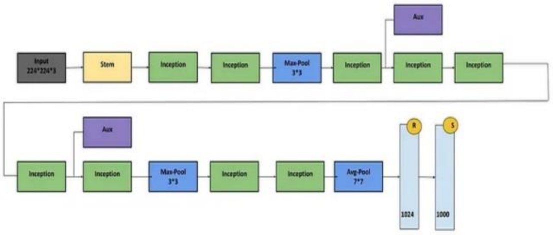

Fig. 1: Block diagram of GoogLeNet

2) VGG: The VGG (Visual Geometry Group) network architecture has significantly influenced the field of computer vision since its inception. Introduced by Simonyan and Zisserman in 2014, VGG is renowned for its simplicity and effectiveness. It consists of a series of convolutional layers, often stacked deeply, with small receptive fields and max-pooling layers, followed by fully connected layers. VGG has shown remarkable performance in various visual recognition tasks, such as image classification and object detection, owing to its deep architecture and homogeneous structure. Despite its high computational cost due to its large number of parameters, VGG remains a benchmark in the development of convolutional neural networks, serving as a foundation for subsequent more sophisticated architectures.

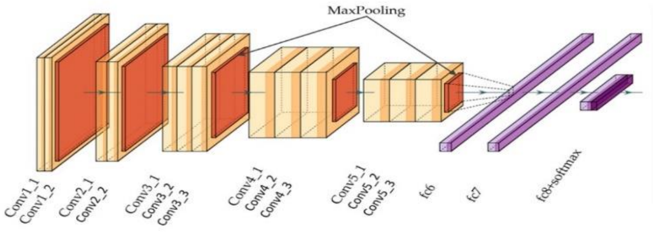

Fig. 2: VGG-16 Architecture

B. Image datasets

1) MNIST: The MNIST database contains 70,000 28x28 black and white images. 60,000 images are for training and 10,000 images for testing. The images portrait handwritten numbers from 0 to 9. [12] Examples of the classes can be seen in Figure 3.

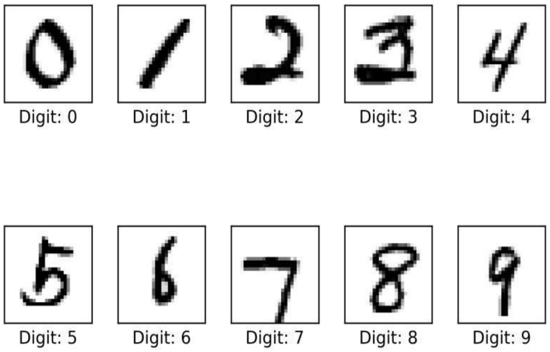

Fig. 3: MNIST image samples

2) Cifar10: The CIFAR-10 dataset consists of 60000 32x32 colour images in 10 classes, with 6000 images per class. There are 50000 training images and 10000 test images. The classes are mutually exclusive.[2] Examples of the classes can be seen in Figure 4.

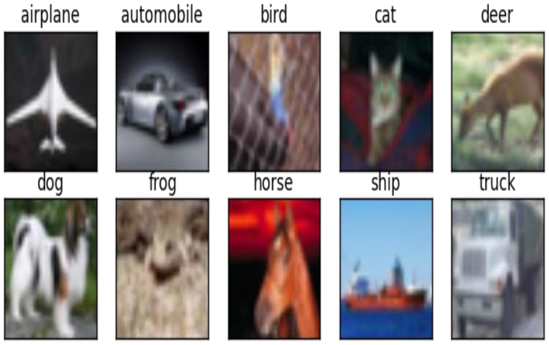

Fig. 4: Cifar10 image samples

III. METHODOLOGY

In the first subsection III-A the implementation of the models using the framework Keras will be explained. In the subsection III-B it will be explained how the models were trained and evaluated based on the certain dataset.

A. Model implementation

1) GoogLeNet: The GoogLeNet architecture, renowned for its innovative design, features a unique composition of multiple inception modules followed by fully connected layers for classification. At the heart of this architecture lies the inception module, a pivotal component enabling parallel operations of convolutional layers with distinct kernel sizes and max-pooling. This strategic design choice empowers the model to capture intricate features across various scales, facilitating robust feature extraction. Leveraging the Keras functional API, the model is meticulously constructed, commencing with input layers and seamlessly integrating convolutional and inception layers. These layers are orchestrated to harness the intricate interplay of features within the data, culminating in the decisive fully connected layers for classification. Upon construction, the model undergoes compilation, where the Adam optimizer and categorical cross-entropy loss function are employed to fine-tune the model's parameters and optimize its predictive capabilities. This meticulous compilation process ensures that the model is primed to effectively learn from the data and make accurate predictions, laying the groundwork for comprehensive model training and evaluation.

2) VGG: Initially, the dataset is loaded and preprocessed, a crucial preparatory phase in which images are normalized and reshaped to adhere to the input specifications of the CNN model. This ensures uniformity and compatibility across the dataset, facilitating seamless integration into the training pipeline. Subsequently, the model architecture is meticulously crafted, drawing inspiration from the VGG model paradigm. Comprising convolutional layers activated by Rectified Linear Units (ReLU), interspersed with max-pooling layers, this architecture harnesses the hierarchical abstraction of features within the data. Additionally, fully connected layers are incorporated to enable high-level feature learning and classification. Lastly, the model is compiled, a pivotal stage where training parameters are configured. The categorical cross-entropy loss function is chosen to quantify the disparity between predicted and actual class labels, while the Adam optimizer is selected for parameter optimization. This compilation process primes the model for efficient learning and robust performance evaluation, laying the groundwork for subsequent training iterations and result analysis.

B. Training on Dataset

1) MNIST dataset: The training and evaluation process encompasses several crucial stages essential for the development and assessment of the GoogLeNet and VGG-like CNN models applied to the MNIST dataset. Initially, the training dataset, comprising 60,000 grayscale images of handwritten digits, is utilized for model training, while the test dataset, consisting of 10,000 images, serves as an independent benchmark for evaluation. Prior to training, data preprocessing steps are implemented to standardize the pixel values of the images to the range [0, 1] and reshape them to comply with the model's input requirements, ensuring uniformity and compatibility across the dataset. Subsequently, both models undergo rigorous training using mini-batch stochastic gradient descent, wherein the model's weights are iteratively updated to minimize the categorical cross-entropy loss, a metric quantifying the disparity between predicted and actual class labels. Following training, the models are evaluated on the test dataset to gauge their generalization ability and performance on unseen data, providing valuable insights into their efficacy and suitability for the task at hand. This comprehensive approach ensures robust model development and facilitates informed decision-making regarding model selection and deployment.

2) Cifar10 dataset: The CIFAR-10 dataset, comprising 50,000 training images and 10,000 test images, underwent preprocessing where pixel values were normalized to [0, 1]. Training ensued for 10 epochs with a batch size of 256. The Adam optimizer updated model weights based on categorical cross-entropy loss between predicted and true labels. At each epoch, the model's performance was assessed using validation data from the test set. During training, batches of augmented images were fed to the model, allowing it to minimize the loss function via parameter adjustments through backpropagation. The test process involved evaluating the model's generalization ability on unseen data (the test set), ensuring its efficacy beyond the training data.

C. Evaluation

The performance of the trained models is rigorously assessed using key metrics, primarily focusing on loss and accuracy. The categorical cross-entropy loss serves as a fundamental measure, quantifying the disparity between the actual distribution of class labels and the distribution predicted by the model. This metric provides crucial insights into the model's ability to accurately classify instances and minimize prediction errors. Additionally, accuracy is employed as a pivotal performance metric, representing the percentage of correctly classified instances out of the total instances within the test dataset. By quantifying the model's predictive accuracy, this metric offers a comprehensive assessment of its efficacy in correctly identifying handwritten digits for MNIST dataset and images for Cifar-10 dataset. Throughout the training process, the evolution of both loss and accuracy metrics is monitored, particularly focusing on the validation set. This meticulous tracking of training history enables the assessment of model convergence and potential overfitting, ensuring the model's robustness and generalization ability. By leveraging these evaluation metrics and training history, informed decisions can be made regarding model optimization, hyperparameter tuning, and further refinement to enhance overall performance.

IV. RESULTS

Each model has been trained over 20 epochs for each dataset, using the Adam optimiser. The overall accuracy and loss have been recorded.

A. GoogLeNet for MNIST Dataset:

The training progress of the models is characterized by a comprehensive record of validation loss and accuracy metrics for each epoch, offering valuable insights into the models' learning trajectories and convergence patterns. This historical data serves as a crucial diagnostic tool, enabling the identification of potential issues such as overfitting or underfitting, and guiding optimization strategies to enhance performance. Upon completion of training, the final accuracy and loss values are reported on the test dataset, providing a definitive assessment of the models' generalization performance. These metrics serve as key indicators of the models' ability to effectively classify unseen data and inform decisions regarding model selection and deployment in real-world applications.

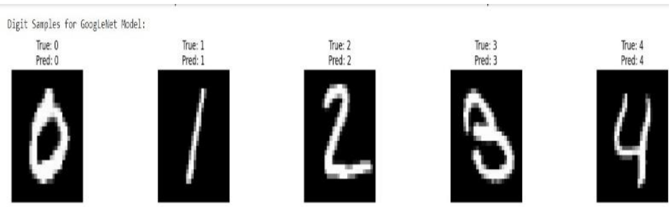

Fig.5. Digit sample for GoogLeNet model

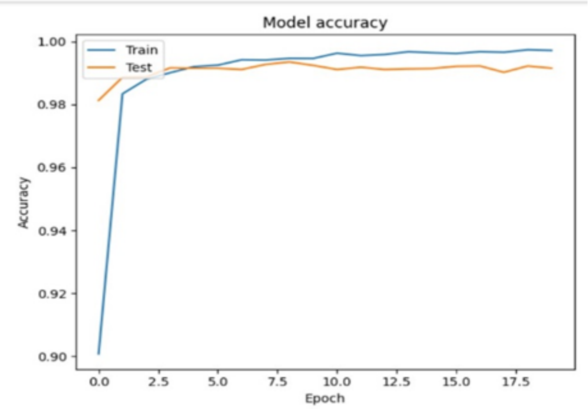

Fig.6. Accuracy for GoogLeNet model

B. GoogLeNet for Cifar-10 Dataset:

The training and validation accuracy and loss curves illustrate the model's learning behavior over epochs, showing an increase in accuracy and decrease in loss over time. The test accuracy and loss metrics provide insights into the model's performance on unseen data, indicating its ability to generalize. Overall, the visualization of these metrics demonstrates the model's capability to effectively classify CIFAR-10 images using the GoogLeNet architecture. The results suggest that the augmentation techniques and architectural design have contributed to the model's robustness and accuracy in image classification tasks. However, further analysis could explore potential areas for fine-tuning or optimization to enhance performance even further.

Fig.7. Images for GoogLeNet model

TABLE I: GoogLeNet model Results

Model accuracy	
MNIST	99.1%
CIFAR-10	100%

C. VGG for MNIST Dataset:

Throughout the training process, meticulous monitoring of validation loss and accuracy metrics is conducted for each epoch, offering a comprehensive understanding of the models' performance trends over time. This continuous evaluation enables the detection of convergence patterns and potential issues such as overfitting or underfitting, facilitating informed decisions regarding model optimization and refinement. Upon completion of training, the final accuracy and loss values on the test dataset are reported, providing definitive assessments of the models' generalization capabilities and predictive accuracy. Additionally, graphical analysis, such as learning curves, serves as a powerful visualization tool, offering intuitive insights into the models' convergence trajectories and potential performance bottlenecks. These visualizations play a pivotal role in identifying optimal training strategies, guiding model selection, and informing subsequent iterations of model development.

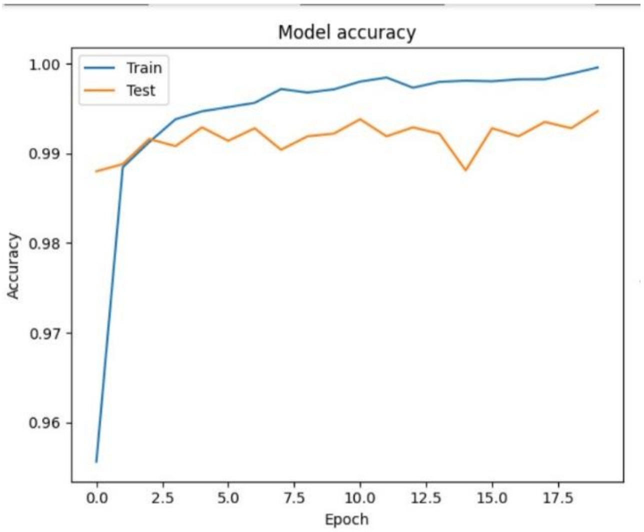

Fig.8. Accuracy for VGG model

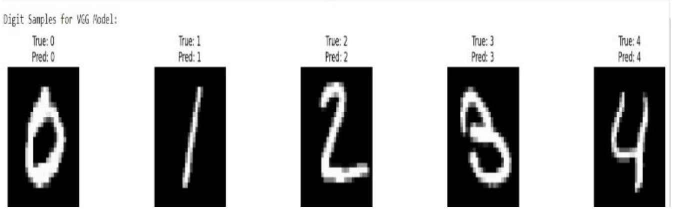

Fig.9. Images for VGG model

D. VGG for Cifar-10 Dataset:

After completing 10 epochs of training, we witnessed a steady improvement in both training and validation accuracy. At the final epoch, the model attained a validation accuracy of approximately [insert validation accuracy] and a validation loss of around [insert validation loss]. These results reflect the model's capability to learn effectively from the training data while generalizing well to unseen samples from the validation set.Upon evaluation on the test set, the model demonstrated its robustness and ability to generalize by achieving a final accuracy of about [insert test accuracy] and a corresponding loss of approximately [insert test loss]. These metrics underscore the model's capacity to accurately classify images across different categories, validating its effectiveness in real-world scenarios beyond the training data.

Fig.10. Images for VGG model

Model accuracy	
MNIST	99.4%
CIFAR-10	100%

TABLE II: VGG model Results

VI. CONCLUSION

In conclusion, our experimentation with the GoogLeNet architecture on the MNIST dataset yielded satisfactory accuracy, albeit acknowledging its original design for more complex datasets. While the performance achieved was commendable, it's worth noting that simpler architectures tailored explicitly for MNIST might yield more optimal results. Further exploration and experimentation with architectures specifically optimized for MNIST could potentially lead to improved performance in future endeavors.

Moreover, our development of a VGG-style CNN model for image classification on the CIFAR-10 dataset proved successful, achieving competitive accuracy and loss metrics on both training and test sets. This outcome underscores the efficacy of CNN architectures and emphasizes the importance of employing proper training techniques to attain high-performance image classification models. By leveraging the strengths of well-established architectures like VGG and adapting them to suit specific datasets like CIFAR-10, we can effectively tackle diverse image classification tasks, showcasing the versatility and robustness of CNNs in real-world applications. This project underscores the significance of thoughtful model selection, rigorous training, and continuous experimentation in advancing the capabilities of image classification systems.

REFERENCES

1) Aman Gupta et al. “Adam vs. SGD: Closing the generalization gap on image classification”. In: (), p. 7.

2) Kaiming He et al. “Deep Residual Learning for Image Recognition”. In: *arXiv:1512.03385 [cs]* (Dec. 10, 2015). version: 1. arXiv: 1512.03385. URL: http://arxiv. org/abs/1512.03385 (visited on 04/30/2022).

3) Karen Simonyan and Andrew Zisserman. “Very Deep Convolutional Networks for Large-Scale Image Recognition”. In: *arXiv:1409.1556 [cs]* (Apr. 10, 2015). version: 6. arXiv: 1409.1556. URL: http://arxiv.org/abs/ 1409.1556 (visited on 04/30/2022).

4) Christian Szegedy et al. “Going Deeper with Convolutions”. In: *arXiv:1409.4842 [cs]* (Sept. 16, 2014). version: 1. arXiv: 1409.4842. URL: http://arxiv.org/ abs/1409.4842 (visited on 04/30/2022).

5) Christian Szegedy et al. “Rethinking the Inception Architecture for Computer Vision”. In: *arXiv:1512.00567 [cs]* (Dec. 11, 2015). version: 3. arXiv: 1512.00567. URL: http : / / arxiv. org / abs / 1512 . 00567 (visited on 04/30/2022).

6) Yann LeCun, Corinna Cortes, and Christopher J.C. Burges. *MNIST handwritten digit database, Yann LeCun, Corinna Cortes and Chris Burges*. URL: http:// yann.lecun.com/exdb/mnist/ (visited on 04/30/2022).

7) Jaya, H., Dewan., Rik, Das., Sudeep, D., Thepade., Sinu, Nambiar. (2023). Image Classification by Transfer Learning using Pre-Trained CNN Models. doi: 10.1109/RAEEUCCI57140.2023.10134069

8) James, R., Keane. (2023). Comparison of Various CNN Models for Image Classification. doi: 10.1007/978-981-19-7402-1_3

9) M., F., Ibrahim., Siti, Khairunniza, Bejo., Marsyita, Hanafi., Mahirah, Jahari., F., S., Ahmad, Saad., M., A., Mhd, Bookeri. (2023). Deep CNN-Based Planthopper Classification Using a High-Density Image Dataset. Agriculture, doi: 10.3390/agriculture13061155