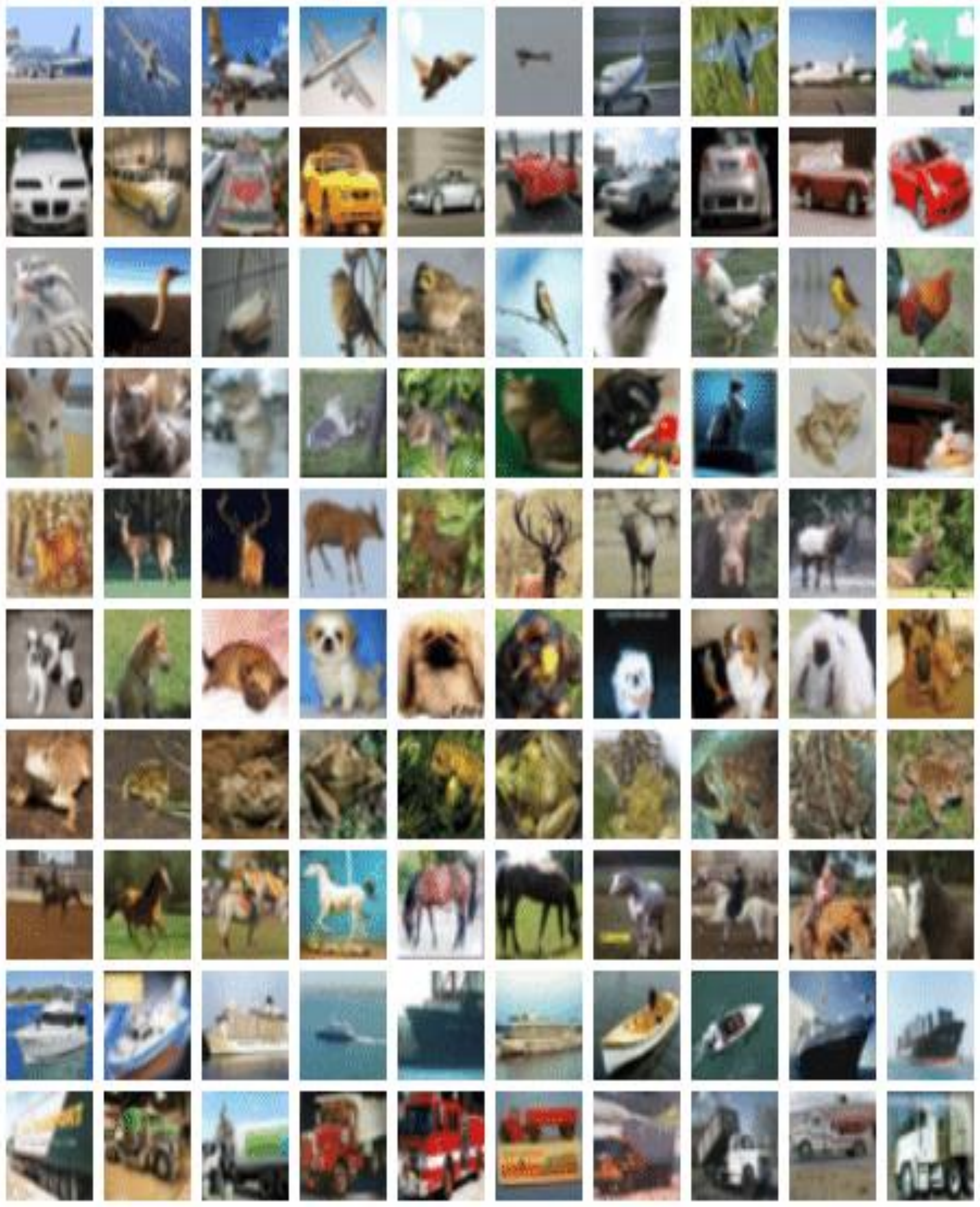
Comparative Analysis of CNN and Transfer Learning Models on CIFAR-10 Dataset

Exploring the potential of transfer learning in image classification using Keras.

[Google Colab Link]

Seyi Falope

Student No: 22023879



INTRODUCTION

In this project, I delve into the field of transfer learning for image classification tasks. Transfer learning a powerful approach that involves reusing a pre-trained model's knowledge, particularly the weights for a new related task. This project focuses on unraveling the principles and significance of transfer learning in image classification, with a keen eye on its implementation within the Keras framework.

The CIFAR-10 dataset developed by the Canadian Institute for Advanced Research will be used in evaluating the effectiveness of transfer learning in comparison to the traditional method of training a model from scratch. CIFAR-10 is a collection of 60,000 32x32 color images splitted 50000 trained set and 10000 test set and distributed among 10 classes, portraying a diverse array of objects and scenes, including airplanes, birds, cats, and more. This meticulous dataset provides a robust challenge for image classification tasks using transfer learning.

This project will be hinged on the following key points

- Overview of transfer learning and its importance in machine learning
- Selection of the pre-trained model used in the project and its original purpose.
- Fine-tuning steps employed to the pretrained model
- Comparison of results: Transfer learning vs. building a model from scratch.
- Limitations and potential areas of improvement.

Transfer Learning And Its Importance In Machine Learning

Transfer learning stands as a cornerstone in the field of machine learning, revolutionizing ho new tasks are approached. Its significance lies in the ability to leverage knowledge gained from one domain and apply it to a different, yet related, domain. In image classification, this involves leveraging pre-trained models' knowledge, such as VGG16, Inception, MobileNet or ResNet, to enhance performance on a different dataset.

Application:

Transfer learning finds its application across various domains, from image and speech recognition to natural language processing. Its versatility allows models to tackle new challenges with a head start, saving computational resources and time. Furthermore, this adaptability fosters a more efficient and accelerated learning process, showcasing the dynamic modification capabilities of transfer learning for diverse tasks

Advantages:

The benefits of transfer learning are manifold. It enables improved model performance, especially in scenarios where labeled data is limited. By building upon pre-existing knowledge, models can generalize better, adapt faster, and achieve higher accuracy. Additionally, this approach serves as a valuable ally in overcoming data scarcity challenges, offering a pathway to enhanced performance even in resource-constrained environments.

Modifications:

Two common modification in transfer learning include fine-tuning and feature extraction. Fine-tuning allows us to adjust specific layers of a pre-trained model to align with the complexity of the new task at hand. Feature extraction involves utilizing the knowledge learned by the pre-trained model's early layers without any changes as a fixed feature extractor for the new task. These modifications exemplify the flexibility of transfer learning, offering tailored adjustments to optimize model performance based on the unique demands of the target task.

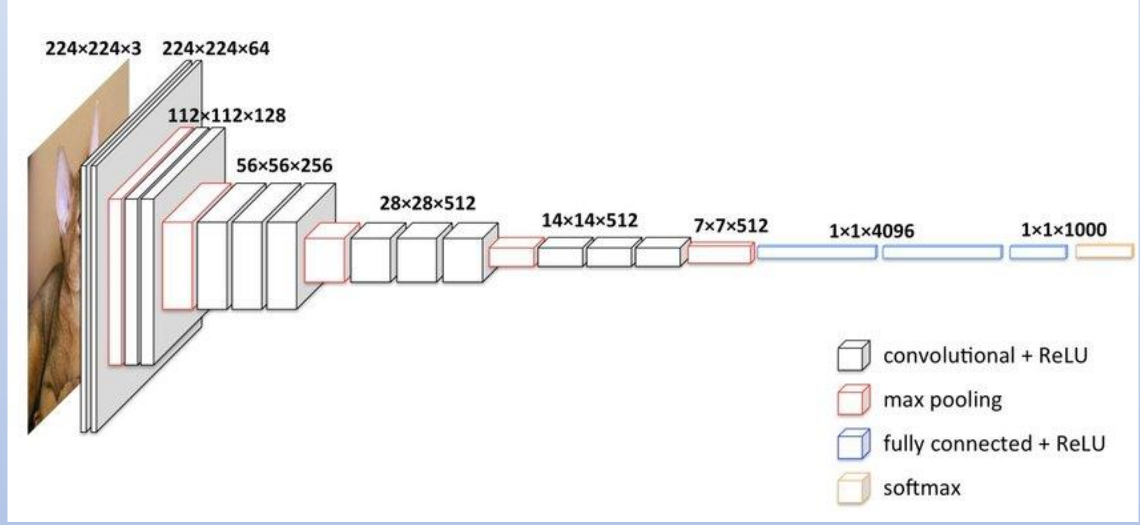
VGG16- A Strategic Choice for Image Classification

My pretrained model of choice for this image classification project is VGG16 (Visual Geometry Group 16) a convolutional neural network (CNN) architecture that originally gained prominence through its exceptional performance in the ImageNet classification challenge. With a sophisticated architecture comprising 16 layers, all featuring 3x3 filters, VGG16 combines simplicity with effectiveness, rendering it an optimal candidate for many transfer learning endeavors.

Purpose and Utility: The genesis of VGG16 lies in its outstanding performance in the ImageNet competition, where it was meticulously trained on over 1.2 million labeled images spanning thousands of categories. VGG16 excels in object detection and classification, showcasing its capability to classify 1000 images across diverse categories with an impressive accuracy of 92.7%.It has become a staple in image classification algorithms, particularly favored for its seamless integration with transfer learning.

In this project, I will be leveraging VGG16 as the pre-trained model bringing forth its wealth of knowledge acquired from the ImageNet challenge. By reusing this pre-existing knowledge and weights, VGG16 model can efficiently learn and adapt to the Cifar10 dataset

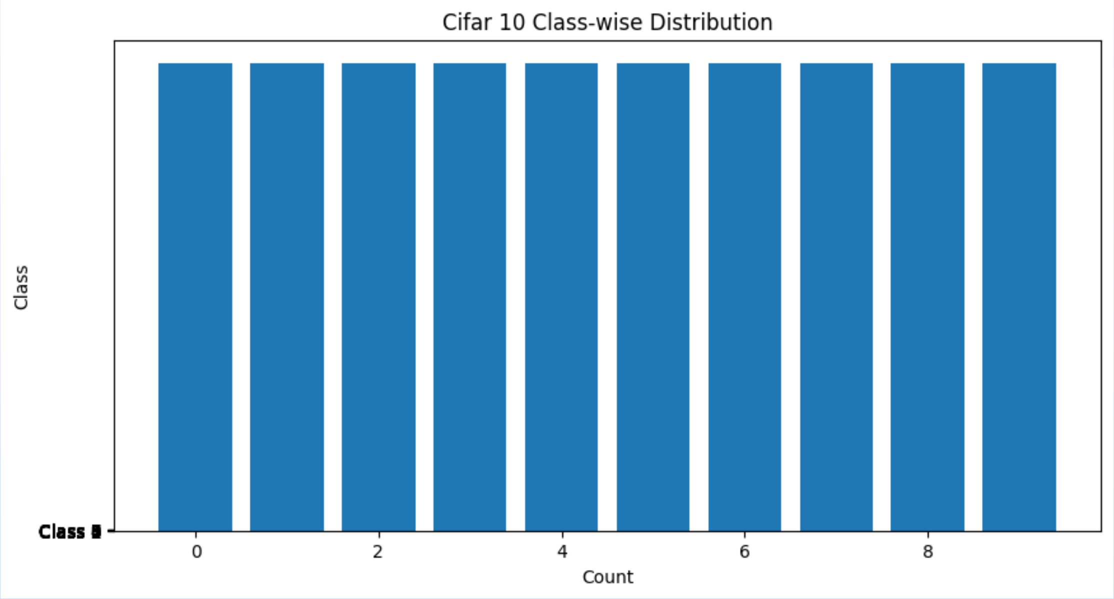
Fig1: Visual Representation of VGG16



Source:ResearchGate

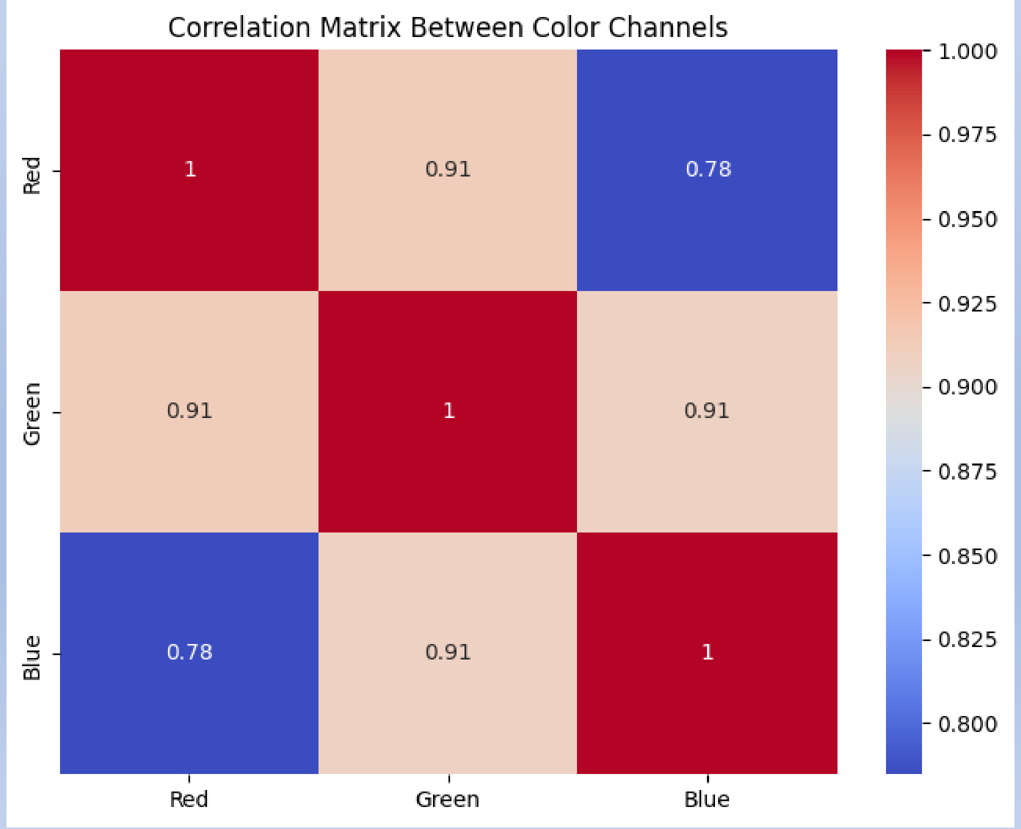
Exploratory Data Analysis (EDA)

Before diving into building the models, an exploratory data analysis was conducted. This step is crucial to have a comprehensive understanding of the dataset, aiding in effective preprocessing, model selection, and potential identification of challenges that may affect model performance.



Class Distribution Analysis:

The class distribution for the CIFAR-10 dataset appears to be uniform across the different classes, indicating a balanced dataset. This is ideal for model training as it prevents the model from developing a bias toward more frequently represented classes. With such even distribution, each class is equally represented, allowing for a fair comparison of model performance across classes.



Heatmap of Color Channel Correlations in CIFAR-10 Dataset

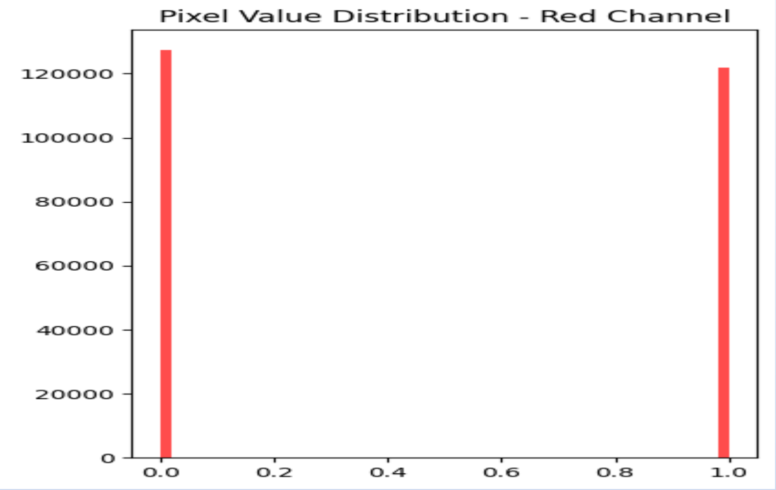
The heatmap displays the correlation coefficients between the red, green, and blue color channels of the CIFAR-10 dataset. The red and green channels show a high correlation of 0.91, indicating a strong linear relationship, suggesting that these channels share a significant amount of information. Similarly, the green and blue channels also have a high correlation of 0.91. The red and blue channels have a slightly lower correlation of 0.78, which is still a strong correlation but indicates a lesser degree of redundancy compared to the red-green and green-blue channel pairs. This high inter-channel correlation suggests that there might be potential for dimensionality reduction or color space transformation in preprocessing steps for machine learning models

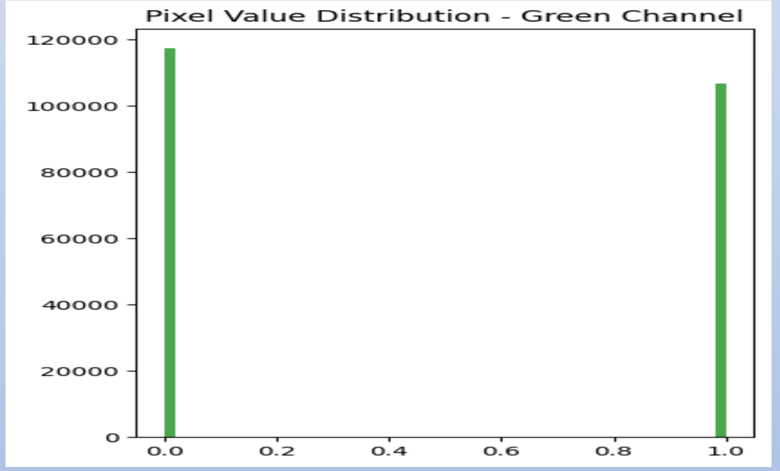
Exploratory Data Analysis (EDA) Cont'd

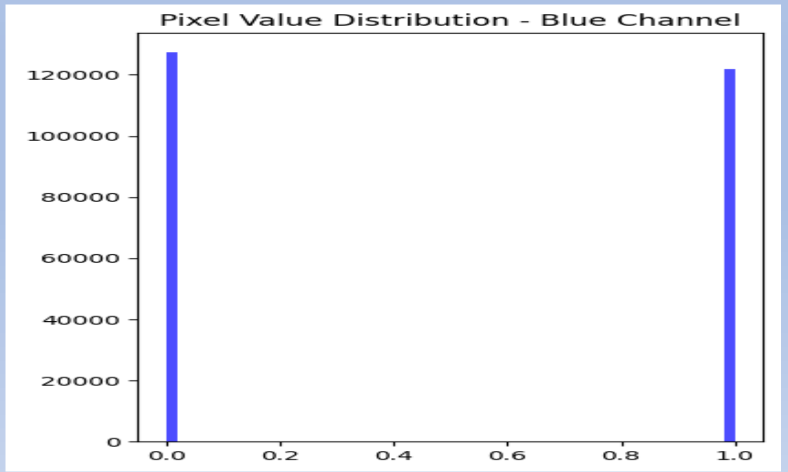
Color Channel Pixel Intensity Distribution:

The histograms depict pixel value distributions for the red, green, and blue channels in the CIFAR-10 dataset, with two predominant spikes near the values 0 and 1. This bimodal distribution suggests that there's a significant number of pixels at the extremes of the intensity spectrum, indicating that many images have regions of both high saturation (likely the subjects of the images) and low saturation (likely the backgrounds). The lack of intermediate values could imply that the images are generally of high contrast, with less gradient or transition between colors. This characteristic might affect certain image processing techniques or the training of machine learning models, as there may be less nuanced color information to distinguish between different classes.

The skewed distribution towards the extremes necessitates careful normalization to ensure that intermediate values are not lost during the scaling process.







Fine-tuning Steps Employed

Unfreezing Layers To Enable The Model capture more task-specific features Cifar10 Dataset

In the fine-tuning phase of this study, I employed a nuanced approach to adapt the pre-trained VGG16 model to the CIFAR-10 dataset. To harness the power of transfer learning, I selectively froze the initial convolutional layers of the VGG16 architecture. By doing so, I retained the pre-trained knowledge embedded in these layers, which had originally learned rich hierarchical features from the vast ImageNet dataset.

Conversely, I strategically unfroze and made trainable not only the dense layers but also the last 4 convolutional layers of the VGG16 model. These unfrozen layers, including both the dense layers responsible for high-level feature representation. These fine tuning step was chosen to capture more task-specific features . Allowing these layers to be fine-tuned facilitated the adaptation of the model's learned representations to the specific characteristics and nuances present in the CIFAR-10 dataset.

This fine-tuning strategy was a deliberate choice to strike a balance between leveraging the generalization power of pre-trained convolutional features and adapting the model's higher-level representations for optimal performance in classifying CIFAR-10 images. By unfreezing the top four convolutional layers, we enabled the model to adapt and fine-tune these higher-level features, creating a tailored model capable of achieving superior results on our targeted image classification task.

Model Preparation and Optimization Techniques

1 **One-Hot Encoding of Labels:**

I converted the class labels into one-hot encoded vectors, which is necessary for the model to perform multi-class classification. This process converts numerical class labels into a binary matrix representation.

2 **Normalization**

I normalized the image data to have pixel values between 0 and 1, which is achieved by dividing the pixel values by the maximum value of 255. This normalization facilitates the model's learning process.

3 **Data Augmentation**

To increase the model's ability to generalize, I implemented data augmentation on the training images. This included applying random transformations such as rotations, shifts, and flips, effectively increasing the dataset size and variability.

4 **Validation Set Allocation**

I allocated 20% of the training data as a validation set. This separate dataset allows me to monitor the model's performance on unseen data, helping me to fine-tune the model and avoid overfitting.

5 **Optimization**

For optimizing the model, I chose the Stochastic Gradient Descent (SGD) optimizer as this resulted in higher model performance of the pretrained model . SGD is a fundamental optimization algorithm that updates model weights based on the gradient of the loss function with respect to the weights, typically using a fixed learning rate and often including momentum to improve convergence on noisy problems.

Model Summary :Transfer Learning Vs. Building A Model From Scratch

MODEL FROM SCRATCH

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 32, 32, 64)        1792

 max_pooling2d (MaxPooling2  (None, 16, 16, 64)        0
 D)

 conv2d_1 (Conv2D)           (None, 14, 14, 32)        18464

 max_pooling2d_1 (MaxPoolin  (None, 7, 7, 32)          0
 g2D)

 dropout (Dropout)           (None, 7, 7, 32)          0

 flatten (Flatten)           (None, 1568)              0

 dense (Dense)               (None, 256)               401664

 dropout_1 (Dropout)         (None, 256)               0

 dense_1 (Dense)             (None, 10)                2570

=================================================================
Total params: 424490 (1.62 MB)
Trainable params: 424490 (1.62 MB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
```

The summary of the model built from the scratch shows a sequential convolutional neural network for consisting of two sets of convolutional and max pooling layers for feature extraction, followed by a flatten layer, a dense layer with dropout for regularization, and a final dense layer for output. It includes dropout layers after max pooling and dense layers to prevent overfitting. The model has a total of 424,490 trainable

TRANSFER LEARNING

```
Model: "sequential_1"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 vgg16 (Functional)          (None, 2, 2, 512)         14714688

 flatten_1 (Flatten)         (None, 2048)              0

 dense_2 (Dense)             (None, 512)               1049088

 dense_3 (Dense)             (None, 256)               131328

 dense_4 (Dense)             (None, 10)                2570

=================================================================
Total params: 15897674 (60.64 MB)
Trainable params: 8818250 (33.64 MB)
Non-trainable params: 7079424 (27.01 MB)
_________________________________________________________________
```

The transfer learning model summary used in this project is a sequential network utilizing the VGG16 architecture as a feature extractor, followed by flattening and dense layers for classification. It includes a dropout layer for regularization before the final classification layer. The model has 8,818,250 trainable parameters and 7,079,424 non trainable parameters

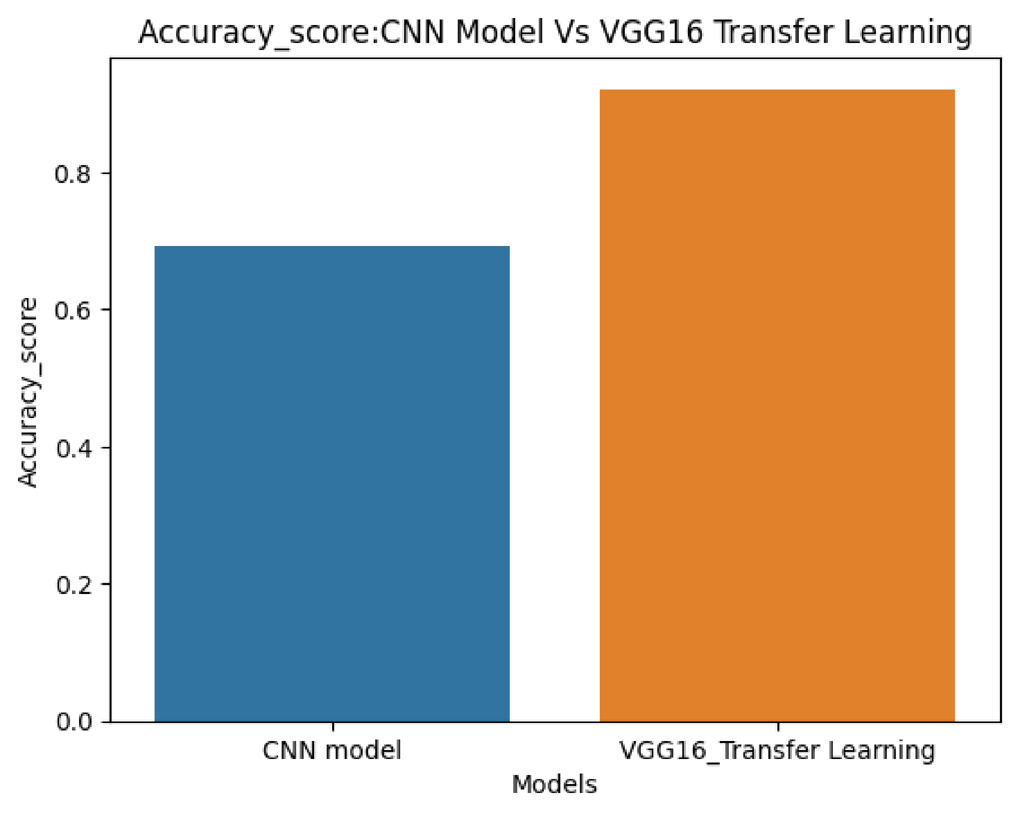
Performance Comparison: Transfer Learning Versus Model from Scratch

The comparison of transfer learning with building a model from scratch on the CIFAR-10 dataset offers clear insights on their performance in image classification , particularly through two key metrics highlighted in this report: accuracy and the confusion matrix.

Accuracy Comparison:

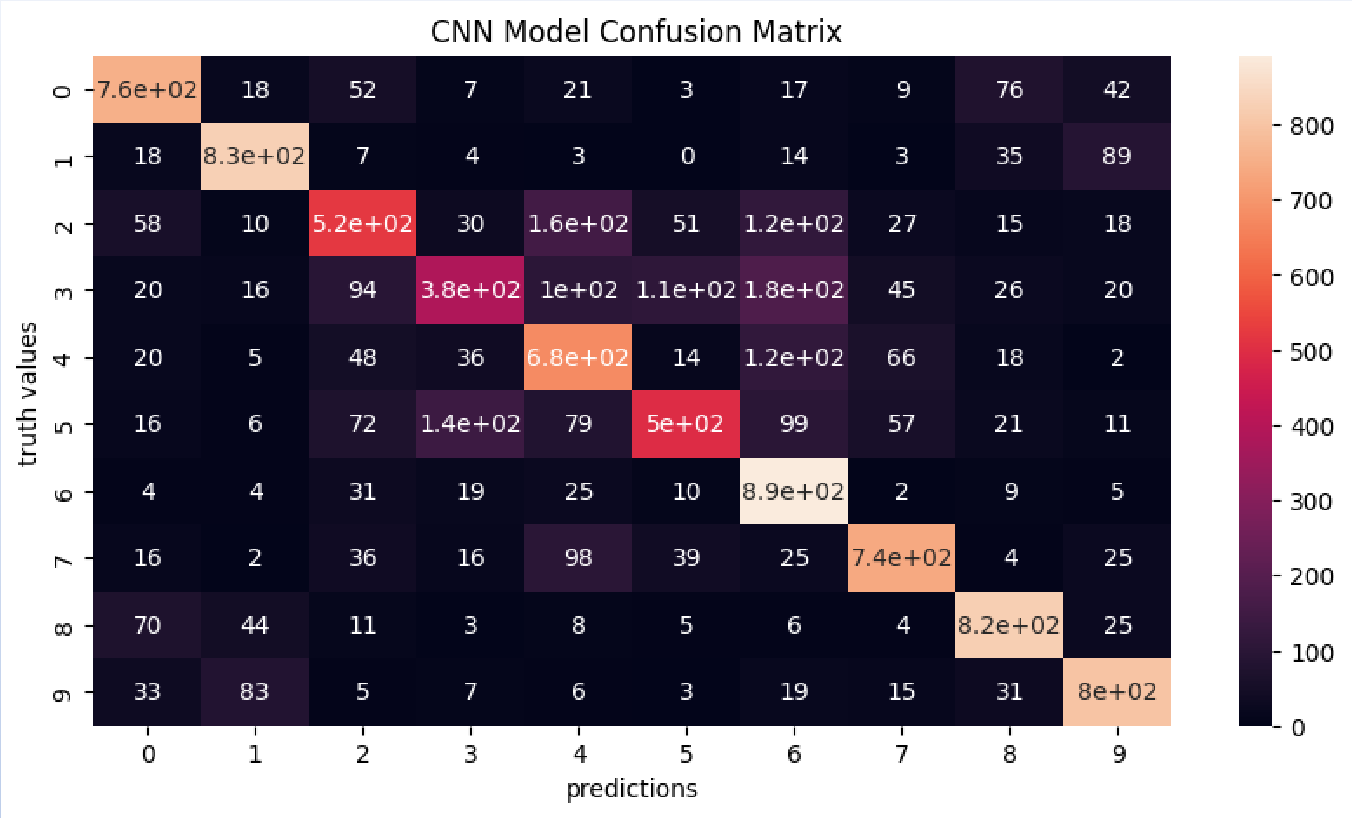
The CNN model from the scratch, depicted in blue, reached an accuracy of 69.7% on the test data. This is quite an impressive outcome for a model built from the ground up, especially on a balanced dataset like CIFAR-10.

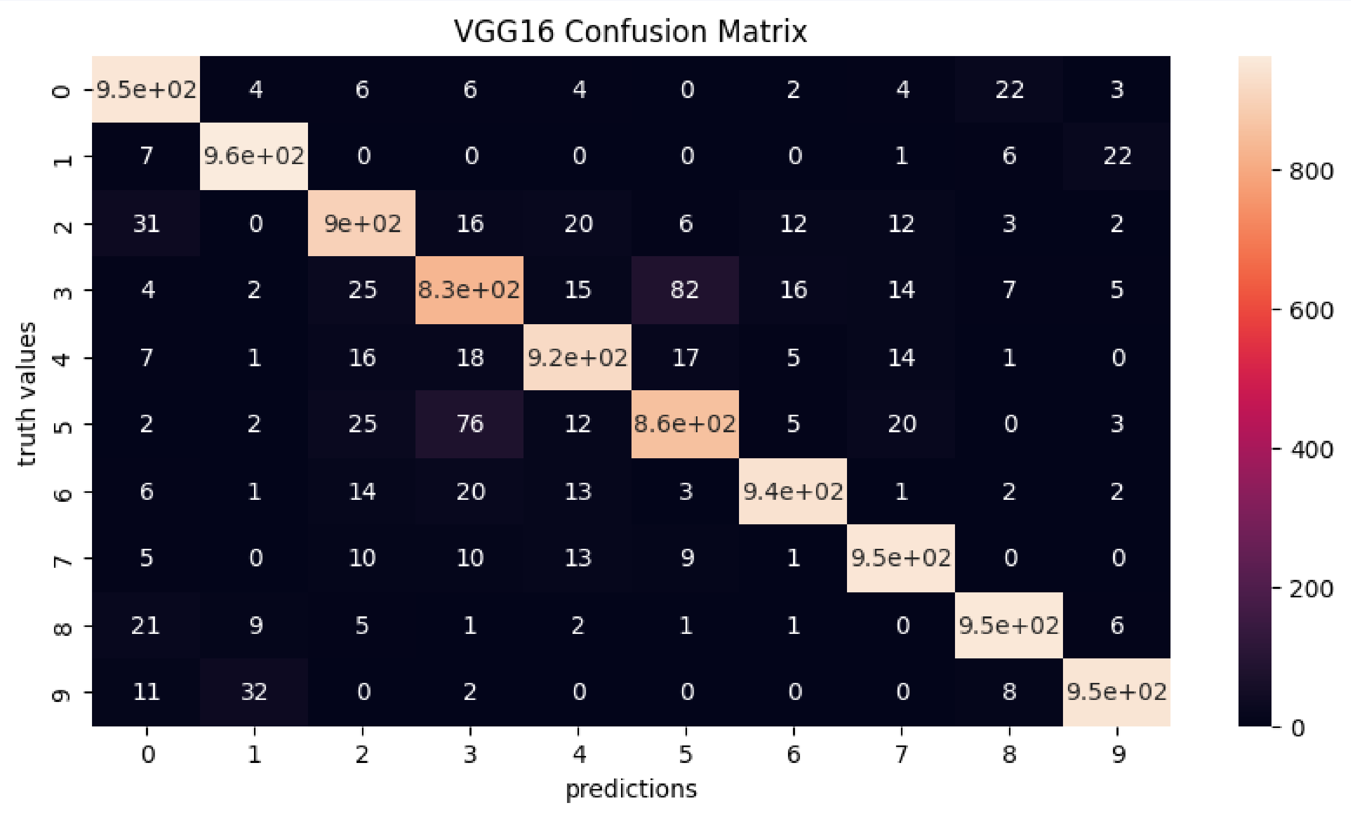
The VGG16 Transfer Learning Model, represented in orange, achieved a notably higher accuracy of 92% on the test data. This demonstrates the effectiveness of transfer learning, where leveraging models pre-trained on extensive datasets results in a substantial boost in performance. The higher accuracy of the transfer learning model can be attributed to the vast and diverse image dataset knowledge it gained from prior training. This advantage enables the model to more effectively recognize and generalize features compared to the CNN model built from scratch



While accuracy provides a clear measure of performance, it lacks the depth to thoroughly analyze the model's specific strengths and weaknesses. Therefore, a confusion matrix was generated for each model. This matrix offers insights into class-specific performance and helps to understand the types of errors each model is making, offering a more detailed view of their capabilities.

Confusion Matrix :Transfer Learning Vs. Model From Scratch





The confusion matrices provide a detailed comparison of the classification performance for our CNN Model and the VGG16 Transfer Learning Model.

The CNN model displays more misclassification across classes, with notable challenges in accurately predicting certain classes such as 2,3,4,5,and 6, suggesting room for improvement in feature extraction and class discrimination.

The VGG16 model shows a stronger diagonal true positives concentration in the confusion matrix, indicating superior classification accuracy, especially for classes 2,3,4,5,6 where the CNN had challenges with. Overall, the VGG16 model outperforms the CNN model, benefiting from advanced feature detectors learned from extensive pre-training phase

Despite, the VGG16 model demonstrating clear advantage and superiority over the CNN model, both models exhibit potential for further refinement to enhance class-specific accuracy."

Conclusion

This project meticulously compared a CNN model built from scratch with a VGG16-based transfer learning model on CIFAR-10. The VGG16 model showcased superior classification accuracy, evident in both the accuracy metrics and the concentrated true positives in the confusion matrix. This success is largely attributed to the intricate feature extraction capabilities embedded in VGG16, pre-trained on an extensive image dataset, underscoring the effectiveness of transfer learning in tasks where deep feature hierarchies play a pivotal role.

However, it's crucial to acknowledge the limitations identified in this analysis:

1. The analysis was conducted on a balanced dataset, potentially limiting its direct applicability to real-world scenarios with imbalanced data.
2. Limited computational resources constrained the depth of hyperparameter tuning and exploration of sophisticated model architectures.
3. The generalizability of transfer learning to significantly different tasks or datasets may necessitate more extensive retraining.

To further enhance model performance, several strategic avenues for improvement can be explored in future endeavors:

1. Data Augmentation: Implementing diverse augmentation techniques to enrich training data.
2. Hyperparameter Tuning: Conducting an extensive search for optimal model parameters.
3. Fine-tuning: Retraining and unfreezing additional layers of the VGG16 model for CIFAR-10 specific adaptation.
4. Ensemble Learning: Leveraging multiple models to reinforce prediction accuracy.

References

1. Pan, S. J., Yang, Q., & et al. (2011). Domain Adaptation via Transfer Component Analysis. IEEE Transactions on Neural Networks, 22(2), 199-210.

2. Tzeng, E., Hoffman, J., Darrell, T., & Saenko, K. (2017). Adversarial Discriminative Domain Adaptation. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 7167-7176.

3. Yosinski, J., & et al. (2015). Understanding Neural Networks Through Deep Visualization. arXiv:1506.06579.

4. Oquab, M., Bottou, L., Laptev, I., & Sivic, J. (2014). Learning and Transferring Mid-Level Image Representations Using Convolutional Neural Networks. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 1717-1724.

5. Rusu, A. A., Colmenarejo, S. G., Gulcehre, C., Desjardins, G., Kirkpatrick, J., Pascanu, R., ... & Hadsell, R. (2016). Policy Distillation. arXiv:1511.06295.

6. Zhang, K., Zhang, Z., Li, Z., & Qiao, Y. (2017). Joint Face Detection and Alignment Using Multitask Cascaded Convolutional Networks. IEEE Signal Processing Letters, 23(10), 1499-1503.

7. Zoph, B., Vasudevan, V., Shlens, J., & Le, Q. V. (2018). Learning Transferable Architectures for Scalable Image Recognition. arXiv:1707.07012.